[19]中华人民共和国国家知识产权局

[51]Int. Cl[6]
G06F 15/16
G06F 11/00

[12]发明专利申请公开说明书

[21]申请号 99104983.7

[43]公开日 1999年12月15日

[11]公开号 CN 1238495A

[22]申请日 99.4.9 [21]申请号 99104983.7

[30]优先权

[32]98.5.11 [33]US [31]09/075,629

[71]申请人 国际商业机器公司

地址 美国纽约州

[72]发明人 S·M·雷普斯 J·鲁兹
K·维达蒂

[74]专利代理机构 中国专利代理(香港)有限公司

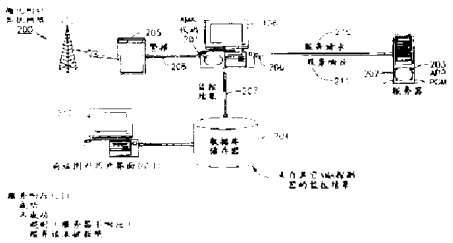
代理人 王 勇 陈景峻

权利要求书7页 说明书30页 附图页数18页

[54]发明名称 分布式计算环境中应用程序可用性和响应的监控与报告

[57]摘要

一种在分布式计算环境中从客户机计算机系统监控服务器计算机系统上驻留的应用程序的性能的方法、系统和程序产品。驻留在客户机计算机的探测器请求应用程序的服务,并根据应用程序的服务响应来记录事务记录。请求和事务记录的生成是由位于客户机计算机的探测器配置数据集控制的。事务记录被提供给中央储存库,统计数据在中央储存库中预处理后插入统计表。显示系统使用户能够交互式地请求和浏览被监控数据的数据集的多个显示画面。



ISSN1008-4274

权　利　要　求　书

1. 在一个有一个服务器计算机与一个客户机计算机相连的计算机网络中，其中服务器计算机通过服务器计算机上的一个应用程序在网络上向客户机计算机提供应用服务，一种在客户机计算机上监控并记录关于由所述应用程序进行的应用服务的性能的数据的方法，该方法的特征在于包含以下步骤:

A. 建立一个参数集供客户机计算机记录关于由所述应用程序进行的应用服务的性能，其中该参数集包括用于反复评估应用程序性能的时间间隔的定义;

B. 从客户机计算机向服务器计算机发出一个服务请求，该服务请求根据所建立的参数集请求由应用程序进行的应用服务的性能;

C. 在客户机计算机接收来自服务器计算机的服务响应并根据所接收的服务响应生成一个事务记录，该事务记录包括关于由所述应用程序进行的应用服务的性能的数据; 以及

D. 按照需要，以所定义的重复时间间隔重复步骤B - C.

2. 根据权利要求1的方法，其中，参数集进一步包括向位于服务器计算机的应用程序发送服务请求所需的数据。

3. 根据权利要求1的方法，其中，参数集包括要发送的服务请求的频率。

4. 根据权利要求1的方法，其中，客户机计算机包括一个定时器，用于确定从发送服务请求到接收服务响应之间的时间。

5. 根据权利要求1的方法，其中，所生成的事务记录包括一个指示从发送服务请求到接收服务响应之间的时间的标志。

6. 根据权利要求 1 的方法，其中的事务记录包括一个表示服务响应是成功的还是不成功的标志，其中的参数集包括一个从发送服务请求到接收服务响应之间的最长时间。

7. 根据权利要求 6 的方法，其中，当所述最长时间被超过时，从客户机计算机向服务器计算机重新发送服务请求。

8. 根据权利要求 7 的方法，其中，当第一次重新发送操作后超过所述最长时间时，可以重复进行发送服务请求的操作，并且其中的

参数集进一步包括一个对重新发送操作的极限，该极限是基于所重复的重新发送的时间间隔或重新发送的操作的次数而重复的。

9. 根据权利要求 8 的方法，其中，如果超过了该极限，就将服务响应记录为不成功的。

10. 根据权利要求 6 的方法，其中，如果应用程序没有响应服务请求提供应用服务，就将服务响应记录为不成功的。

11. 根据权利要求 10 的方法，其中，如果服务响应是不成功的，则应用程序是不可用的，并在所生成的事务记录中包括一个表示应用程序是可用的还是不可用的标志。

12. 根据权利要求 11 的方法，其中，参数集进一步包括在一个确定时期内应用程序可以不可用的最大时间量。

13. 根据权利要求 1 的方法，其中，参数集进一步包括一个时间表，时间表对应的是应用程序何时将可用于响应服务请求的各时间。

14. 根据权利要求 13 的方法，其中的时间表用于选通或禁止服务请求的发送，或者其中所生成的事务记录包括一个表示按照时间表，应用程序是否可用的标志。

15. 根据权利要求 1 的方法，其中的数据集包括一个有关应用服务性能的阈值集，其中该方法进一步包括以下步骤：

分析所生成的事务记录，判断阈值集中是否有任何阈值已经被对事务请求的响应超过；

如果确定阈值集中有的阈值已经被超出，则将一个报警信号发送给一个报警机构。

16. 根据权利要求 15 的方法，进一步包括以下步骤：

分析随后生成的事务记录，判断被超出过的阈值是否不再被超出；

将一个取消信号发送给报警机构，取消上一次发送的报警信号。

17. 根据权利要求15的方法，其中，报警信号包括关于被监控应用程序、服务器计算机和被超出的阈值的数据。

18. 根据权利要求 17 的方法，其中，报警机构用报警信号来联系服务实体。

19. 根据权利要求1的方法，其中，所生成的事务记录被存储在一个储存库中，储存库的位置包含在参数集中。

20. 根据权利要求1的方法，其中，所接收的服务响应与所发送的服务请求相关，并且应用程序生成服务响应时不需要发送服务请求的客户机计算机的验证。

21. 根据权利要求1的方法，其中，由客户机计算机使用、用于记录由应用程序进行的应用服务的性能的参数集，是通过一个从客户机计算机的用户到客户机计算机的用户界面提供的，其中与所接收的服务响应和所生成的事务记录相关的数据是从客户机计算机提供给用户界面供用户浏览的。

22. 在一个有一个服务器计算机与一个客户机计算机相连的计算机网络中，其中服务器计算机通过服务器计算机上的一个应用程序在网络上向客户机计算机提供应用服务，一种在客户机计算机上监控并记录关于由所述应用程序进行的应用服务的性能的数据的装置，该装置的特征在于包含：

用于建立一个参数集供客户机计算机使用用以记录关于由所述应用程序进行的应用服务的性能的装置，其中该参数集可包括用于反复评估应用程序性能的时间间隔的定义；

用于从客户机计算机向服务器计算机发出一个服务请求的装置，该服务请求根据所建立的参数集请求由应用程序进行的应用服务的性能；

用于在客户机计算机接收来自服务器计算机的服务响应的装置以及根据所接收的服务响应生成一个事务记录的装置，该事务记录包括关于由所述应用程序进行的应用服务的性能的数据。

23. 根据权利要求22的装置，其中，参数集进一步包括通过发送装置向位于服务器计算机的应用程序发送服务请求所需的数据。

24. 根据权利要求22的装置，其中，参数集包括要由发送装置发送服务请求的频率。

25. 根据权利要求22的装置，其中，客户机计算机包括一个定时器，用于确定从发送装置发送服务请求到接收装置接收服务响应之间的时间间隔。

26. 根据权利要求22的装置，其中，所生成的事务记录包括一个指示从发送服务请求到接收服务响应之间的时间间隔的标志。

27. 根据权利要求 22 的装置，其中的事务记录包括一个表示服务响应是成功的还是不成功的标志，其中的参数集包括一个从发送装置发送服务请求到接收装置接收服务响应之间的最长时间间隔。

28. 根据权利要求 27 的装置，其中，当所述最长时间间隔被超过时，发送装置从客户机计算机向服务器计算机重新发送服务请求。

29. 根据权利要求 28 的装置，其中，当第一次重新发送操作后超过所述最长时间间隔时，可以重复进行再发送服务请求的操作，并且其中的参数集进一步包括一个对重新发送操作的极限，该极限基于所重复的重新发送的时间间隔或重新发送的操作的次数而进行重复。

30. 根据权利要求 29 的装置，其中，如果超过了该极限，就将服务响应记录为不成功的。

31. 根据权利要求 27 的装置，其中，如果应用程序没有响应服务请求提供应用服务，就将服务响应记录为不成功的。

32. 根据权利要求 31 的装置，其中，如果服务响应是不成功的，则应用程序是不可用的，并在所生成的事务记录中包括一个表示应用程序是可用的还是不可用的标志。

33. 根据权利要求 32 的装置，其中，参数集进一步包括在一个确定时期内应用程序可以不可用的最大时间量。

34. 根据权利要求 22 的装置，其中，参数集进一步包括一个时间表，时间表对应的是应用程序何时将可用于响应服务请求的各时间。

35. 根据权利要求 34 的装置，其中的时间表用于选通或禁止发送装置对服务请求的发送，或者其中所生成的事务记录包括一个表示按照时间表，应用程序是否可用的标志。

36. 根据权利要求 22 的装置，其中的参数集包括一个有关应用服务性能的阈值集，其中该装置进一步包括：

用于分析来自生成装置的事务记录的装置，分析的目的是判断阈值集中是否有任何阈值已经被对事务请求的响应超出；

用于如果确定阈值集中有的阈值已经被超出，则将一个报警信号发送给一个报警机构的装置。

37. 根据权利要求 36 的装置，进一步包括：

用于分析随后生成的事务记录以判断被超出过的阈值是否不再被超出的装置；

用于将一个要求取消上一次发送的报警信号的取消信号发送给报警机构的装置。

38. 根据权利要求 36 的装置，其中，报警信号包括关于被监控的应用程序、服务器计算机和被超出的阈值的数据。

39. 根据权利要求 38 的装置，其中，报警机构用报警信号来联系服务实体。

40. 根据权利要求22的装置，其中，所生成的事务记录被存储在一个储存库中，储存库的位置包含在参数集中。

41. 根据权利要求22的装置，其中，在接收装置接收的服务响应与从发送装置发送的服务请求相关，并且应用程序生成的服务响应时不需要发送服务请求的客户机计算机的验证。

42. 根据权利要求22的装置，其中，由客户机计算机使用、用于记录由应用程序进行的应用服务的性能的参数集，是通过一个从客户机计算机的用户到客户机计算机的用户界面提供的，其中与所接收的服务响应和所生成事务记录相关的数据是从客户机计算机提供给用户界面供用户浏览的。

43. 在一个有一个服务器计算机与一个客户机计算机相连的计算机网络中，其中服务器计算机通过服务器计算机上的一个应用程序在网络上向客户机计算机提供应用服务，一种数字处理设备可读的程序存储器，程序存储器存储着可由数字处理设备执行、在客户机计算机上进行监控并记录关于由应用程序进行的应用服务的性能的数据的方法步骤的指令程序，该方法的特征在于包含以下步骤：

A. 建立一个参数集供客户机计算机使用以记录关于由所述应用程序进行的应用服务的性能，其中该参数集包括用于反复评估应用程序性能的时间间隔的定义；

B. 从客户机计算机向服务器计算机发出一个服务请求，该服务请求根据所建立的参数集请求由应用程序进行的应用服务的性能；

C. 在客户机计算机接收来自服务器计算机的服务响应并根据所接收的服务响应生成一个事务记录，该事务记录包括关于由所述应用程序进行的应用服务的性能的数据；以及

D. 按照需要，以所定义的重复时间间隔重复步骤B－C。

44. 根据权利要求43的程序存储器，其中，参数集进一步包括向位于服务器计算机的应用程序发送服务请求所需的数据。

45. 根据权利要求43的程序存储器，其中，参数集包括发送服务请求的频率。

46. 根据权利要求43的程序存储器，其中，客户机计算机包括一个定时器，用于确定从发送服务请求到接收服务响应之间的时间间隔。

47. 根据权利要求43的程序存储器，其中，所生成的事务记录包括一个指示从发送服务请求到接收服务响应之间的时间间隔的标志。

48. 根据权利要求 43 的程序存储器，其中的事务记录包括一个表示服务响应是成功的还是不成功的标志，其中的参数集包括一个从发送服务请求到接收服务响应之间的最长时间间隔。

49. 根据权利要求 48 的程序存储器，其中，当所述最长时间间隔被超过时，从客户机计算机向服务器计算机重新发送服务请求。

50. 根据权利要求 49 的程序存储器，其中，当第一次重新发送操作后超过所述最长时间间隔时，可以重复进行再发送服务请求的操作，并且其中的参数集进一步包括一个对重新发送操作的极限，该极限基于所重复的重新发送的时间间隔或重新发送的操作的次数而进行重复。

51. 根据权利要求 50 的程序存储器，其中，如果超过了该极限，就将服务响应记录为不成功的。

52. 根据权利要求 48 的程序存储器，其中，如果应用程序没有响应服务请求提供应用服务，就将服务响应记录为不成功的。

53. 根据权利要求 52 的程序存储器，其中，如果服务响应是不成功的，则应用程序是不可用的，并在所生成的事务记录中包括一个表示应用程序是可用的还是不可用的标志。

54. 根据权利要求 53 的程序存储器，其中，参数集进一步包括一个在确定时期内应用程序可以不可用的最大时间量。

55. 根据权利要求 43 的程序存储器，其中，参数集进一步包括一个时间表，时间表对应的是应用程序何时将可用于响应服务请求的各时间。

56. 根据权利要求 55 的程序存储器，其中的时间表用于选通或禁止服务请求的发送，或者其中所生成的事务记录包括一个表示按照时间表，应用程序是否可用的标志。

57. 根据权利要求 43 的程序存储器，其中的数据集包括一个有关应用服务性能的阈值集，其中该方法进一步包括以下步骤：

分析所生成的事务记录，判断阈值集中是否有任何阈值已经被对事务请求的响应超出；

如果确定阈值集中有的阈值已经被超出，则将一个报警信号发送给一个报警机构。

58. 根据权利要求 57 的程序存储器，进一步包括以下步骤：

分析随后生成的事务记录，判断被超出过的阈值是否不再被超出；

将一个取消信号发送给报警机构，取消上一次发送的报警信号。

59. 根据权利要求57的程序存储器，其中，报警信号包括关于被监控的应用程序、服务器计算机和被超出的阈值的数据。

60. 根据权利要求 59 的程序存储器，其中，报警机构用报警信号来联系服务实体。

61. 根据权利要求43的程序存储器，其中，所生成的事务记录被存储在一个储存库中，储存库的位置包含在参数集中。

62. 根据权利要求43的程序存储器，其中，所接收的服务响应与所发送的服务请求相关，并且应用程序生成服务响应时不需要发送服务请求的客户机计算机的验证。

63. 根据权利要求43的程序存储器，其中，由客户机计算机使用、用于记录由应用程序进行的应用服务的性能的参数集，是通过一个从客户机计算机的用户到客户机计算机的用户界面提供的，其中与所接收的服务响应和所生成事务记录相关的数据是从客户机计算机提供给用户界面供用户浏览的。

说　明　书

分布式计算环境中应用程序可用性和响应的监控与报告

本发明的相关参考文献为下列共同未决美国专利申请：《用于分布式计算环境的基于客户机的应用程序可用性和响应监控与报告》（Luzzi 等，S/N 09/076,050），《用于顺序检索和显示多个相关数据集的交互式显示系统》（Luzzi 等，S/N 09/075,704），《用于在网络上建立数据报告和显示通讯的方法、系统和程序产品》（Luzzi 等，受让律师档案号 P0998081）。各申请案都受让给本受让人，与本发明同时提交，本文引用其作为参考。

本发明总体涉及网络系统服务领域，特别是基于最终用户的应用程序可用性和响应的监控与报警系统、方法和程序产品。更具体来说，本发明能够从采用分布式计算环境中应用程序的最终用户的角度，对应用程序的可用性和响应时间或其它期望的性能量度进行监控。此外，本发明还提供一种可随时存取的报告系统，用于动态地通报应用程序监控的实时结果。该监控系统的实现不依赖具体的平台，不影响被监控应用程序的性能，并且容易维护。该报告系统能够通过图形显示，从网络的中央储存库在网络的任何用户可用的各种粒度的层次上对监控结果进行实时分析。此外，还包括建立应用程序的性能阈值的功能，以及判断何时所建阈值受到违反、表明程序性能出现异常的功能。本发明进一步提供向服务或支持实体发送的、表示阈值违反的报警信号，目的是为了迅速地向功能不良的应用程序提供适当的服务。

数据处理系统设计中普遍流行的趋势是采用分布式计算环境，在分布式计算环境中，最终用户在一个或多个各自包括多个互连计算机的互连网络上存取应用程序和数据。典型的分布式计算环境中，最终用户团体所用的桌面计算机或网络计算机，作为各局域网（LAN）上的客户机连接到服务器，后者进而可以本地或远程地连接到其它这种服务器上。例如，一个商业企业可以运行着数个位于不同地理位置的营业处所的互连 LAN。位于一定营业处所的 LAN 服务器之间是相互连

接的，并进一步通过广域网（WAN）互连到远程营业处所的网络中的服务器。

商业企业越来越采用这种计算模型，目的是减少操作、维护和更新独立分割的“分片”计算系统的开支。具有这种分布式计算模型特征的互连网络便利了对应用程序和数据的优先化，即将负有关键使命的应用程序和数据驻留在高端高带宽服务器上，将较不重要的应用程序和数据分配给相对较低端服务器。此外，这种高度地分布的处理模型一般要包含这样的特点，即保证系统即使在单个甚至多个服务器出现故障或进行维护时也能继续正常工作，继续可用。

要实现这样复杂的分布式计算模型，与此同时向用户提供许多益处，也为网络管理者带来了相应复杂的网络管理问题。在互连网络中可以应用异类的操作系统。不同的服务器上可以运行不同的应用程序，以及相同应用程序的不同版本。在网络的本地化或分布式部分发生的故障并不是统一报告的，因此会在很大程度上使修正行动受到推迟。

在很多情况下，企业内部或外部的信息技术服务（IT）组织担负着管理分布式计算环境的责任。一般来说，在与这种代理机构签订的服务水平协议（service level agreement - SLA）中，规定了对于这种网络的用户来说，应用程序可用性和响应时间的期望水平。要履行合同义务，就要求坚持达到这些期望的基准水平；如果不能达到这些基准水平，就会导致客户业务的丧失。因此，能够提供关于应用程序可用性和响应时间的实时数据的应用程序监控系统，对于这种组织来说是一种非常有价值的资产。

已经有许多网络管理工具被开发出来，用于辅助网络管理者对分布式计算系统的性能进行监控。例如，国际商用机器公司（即 IBM，也是本发明的受让人）开发的产品 System Performance Monitor/2 提供了一种图形界面，用于表示处理系统中各种硬件资源的性能。但是该产品并不向最终用户指示软件应用程序的可用性和响应，也不能对监控数据的结果进行深入分析。IBM Netfinity（R）Manager 软件提供对服务器资源以及在客户机层的操作系统资源的网络监控，然而它也是存在于服务器层，并不监控基于客户机的对应用程序的访问。因此，它并不向 IT 专业人员提供为了分析是否达到如上所述、其中

许多都是从网络的最终用户或客户的角度规定的基准水平所需的信息。

有许多无源监控系统是为了从分布式计算系统中的服务器和/或客户机收集可用性数据。

例如，美国专利号 4,858,152 的《操作员对监控应用程序的访问》（发明人是 Estes，1989 年 8 月 15 日授权，转让给本受让人）描述的一种基于微机的监控系统，能同时监控在一个大型计算机上运行的多个主应用程序，能总结监控信息并在微机系统的显示屏上用图形方式显示该信息，还提供一种用于指出达到用户所定义阈值的报警机制。Estes 描述的“多系统应用程序监控器（MSAM）”接收主机现有的总结信息，并缩减信息，使其能精确地描述主机上运行的应用程序。

同样，美国专利号 5,483,468 的《并发地记录和显示系统性能数据的系统与方法》（发明人是 Chen 等人，1996 年 1 月 9 日授权，转让给本受让人）描述了一种在网络上交互选择性能统计数据的性能监控工具。该工具采用一个在服务器上运行的数据供应者守护程序，数据供应者守护程序存储向数据消费者程序选择性地提供的统计数据，数据消费者程序接着再处理出对所需统计数据的报告。Chen 等人的专利的一个优点是，数据消费者程序不必包含任何关于由数据供应者守护程序维持的统计数据的在先信息。Chen 等人的专利提供了一种用于俘获系统数据并记录该数据供以后回放用的机制。

尽管以上所述的各专利能向网络管理者提供有价值的信息，但是它们本身并不测试应用程序可用性或响应时间，而是依靠系统其它部分生成的数据。在 Estes 的专利中，这种信息已经在主机中存在，可用于向微机提供；在 Chen 等人的专利中，系统统计数据是在服务器俘获、然后提供给数据收集器的。所以，这两个专利中的这些监控工具都不生成相关的基于客户机的可用性信息，而是局限于收集和报告业已存在的关于系统性能的信息。如果没有从客户机角度的关于应用程序可用性和响应时间的相关数据可供这些工具使用，这些工具就不会满足数据管理者的目的。

有一些监控系统披露了能独立地生成指示分布式计算系统的状态的信息并收集和报告所生成信息的机制。

美国专利号 5,621,663 的《监控计算机系统的系统与方法》(发明人是 Skagerling,1997 年 4 月 15 日授权,转让给 ICL Systems AB),描述了一种能监控并改变计算机网络的操作的系统,其方法是,修改应用程序,在应用程序中添加一个事件报告生成器,事件报告生成器按照事件处理机中灵活的规则库,向事件处理机通报被监控事件的出现情况,事件处理机将特定事件的出现与预定行动相关联。事件报告生成程序是在系统中运行的应用程序内执行的,它在应用程序执行期间报告预定事件的发生情况。

美国专利号 5,655,081 的《采用智能自动代理结构来监控和管理分布式计算环境中计算机资源和应用程序的系统》(发明人是 Bonnell 等人,1997 年 8 月 5 日授权,转让给 BMC Software,INC.)描述的一种管理应用程序和其它服务器资源的系统中,在网络的每个服务器计算机中安装一个代理程序。所安装的代理程序执行的询问功能,识别它们驻留在哪个系统,什么资源是可用的,并监控服务器上出现的资源和应用程序的方方面面。代理程序与网络上的管理者软件系统通讯,以便能够对整个网络上存在的资源和应用程序及其当前状态进行连续的更新显示。

美国专利号 5,675,798 的《有选择地同时监控多处理服务器中过程的系统与方法》(发明人是 Chang,1997 年 10 月 7 日授权,转让给本受让人)描述的一种监控系统中,关于各客户机应用程序的状态的信息,是按照由服务器过程所反映的那样被获取并提供给网络管理者的。服务器过程监控程序将信息提供给处于客户机-服务器网络内部各客户机过程的粒度层次的网络管理者。

上述各个例子中的监控系统要求在服务器层安装一个入侵型监控程序或探测器—或者安装在如 Skagerling 专利中的服务器上运行的应用程序中,或者安装在管理状态下的服务器上运行的应用程序中,从服务器上运行的被监控应用程序收集信息。无论哪一种情况,探测器的结果都不具有如客户机所体验的那种指导性,因为信息是在网络的服务器侧而不是客户机侧生成和收集的。此外,向网络中运行的服务器添加这种监控程序也带来了维护问题,容易想见,其性质就像各个服务器添加另一个应用程序一样。此外,这些监控程序的执行也会极大地降低它们主服务器的性能,进而降低服务器所服务的网络

的性能，带来这样相互矛盾的结果，即为了达到有效管理网络的目的而采用的工具，恰恰又是导致网络效率不高的原因。

从以上讨论可知，从客户机的角度生成可用性和响应时间信息或任何其它所需要的应用程序度量的新的应用程序监控系统，对网络管理者具有很高的价值。这种系统应当设计成作为在客户机计算机系统可与之相连的复杂分布式计算环境中的任一点执行的探测器，探测器的功能对网络性能的影响应当微不足道。这种系统应当是可按要求定制的，能提供实时报警信号，向接收者警告用户定义阈值的越过，诸如最大可允许响应时间或者被监控应用程序的最小可用性。

监控系统应当提供关于例如应用程序可用性和响应时间的动态报告，报告能被观察者剪裁，以图形或图表形式，显示与该观察者相关的实时或归档监控信息。报告应当以这样的方式来显示，使得浏览者可按图形、图表或其它方式来显示有关许多服务器和/或应用程序的性能的信息，应当提供交互式手段，使浏览者能“下探”信息，以阅览关于特定服务器或应用程序的数据，并且/或者使浏览者能从该数据“上探”信息，阅览更大范围的性能数据。

性能报告应当随时向以任何方式访问网络的任何人开放，报告中的数据应当驻留在网络的中央储存库中，中央储存库中包含与储存数据有关的预先处理过的统计信息。应当通过与网络的有线或无线连接，向人们提供对这种信息的访问，这些人包括网络管理者、咨询人员、网络应用程序的最终用户。

最后，这种系统应当容易实现，维护简便，才能成为网络管理者的辅助而不是进一步的负担。

本发明克服了以上所述的现有技术的问题和缺点，并具有更进一步的优点，其中，描述了用于在分布式计算环境中实现基于客户机的应用程序监控器的方法、装置和程序产品。本发明的优点在于，提供用于监控、报告的技术，以及根据驻留在分布式计算环境中与客户机计算机系统相连的服务器计算机上的应用程序的应用服务的性能，生成基于性能的报警信号的技术。

本发明一个实施例中的计算机网络包括一个服务器计算机，服务器计算机有一个应用程序向连接的客户机计算机系统提供应用服

务，其中，客户机计算机系统通过客户机计算机上驻留的应用探测软件，记录关于应用程序的服务的性能的信息。

在客户机计算机中建立了一个参数集合或探测配置数据，用于记录应用程序的性能。客户机计算机探测器按照这些参数进行配置，向服务器计算机发出服务请求，请求应用程序的应用服务的性能。

相应地，服务器计算机生成一个服务响应，响应可能表示请求正在得到服务（成功响应），或者可能表示请求遭到拒绝（不成功响应）；还有一种响应，实际上就是在预定的超时时间内，服务器系统没有任何响应（不成功响应）。

根据所得到的响应，在客户机计算机生成一个事务记录，该记录含有关于应用程序的应用服务的性能的信息。

最后，这种请求－响应周期按照提供给客户机计算机系统的参数重复进行。

本发明一个实施例中的这些参数包括这类信息，诸如应用程序的名称、服务器系统的地址、访问服务器计算机的频率、应用程序可用性和阈值数据的安排—诸如最小可用性水平和最大响应时间水平，以及不响应的应用程序的超时间隔时间，得到不成功响应时是否重试请求、重试的频率。其它示例性探测器配置数据包括，远程中央储存库表示发送生成的事务记录的标志，以及表示在预定时间段内应用程序不可用时，是中断应用程序监控还是在生成的事务记录中指出这些预定时段的指示。

在本发明的另一个实施例中，当超出这些预定阈值时，会导致客户机计算机生成报警信号，服务器实体收到报警信号后会作出反应，对相关应用程序提出解决问题的办法。

在本发明的一个实施例中，服务请求是一个向客户机计算机提供文件的请求。优选实施例中的服务请求是无须进行客户机身份验证就进行服务的请求。

本发明的另一个实施例中的客户机计算机，包括一个定时机构，用于确定应用程序的响应时间，这个时间由服务请求到服务响应之间的时间间隔决定。在优选实施例中，一个生成的事务记录的事务记录信息，包括应用程序是否成功地响应了请求（即，应用程序是否可用）以及应用程序的响应时间。

根据本发明的一个相关方面，提供一个中央储存库，用于从计算机网络中任何客户机计算机处执行的任何探测器，接收生成的事务记录。

根据探测器的配置，判断是否要将某特定事务记录发送到中央储存库，如果是，就通过数据库加载模块，将该事务记录装入储存库中的一个原始数据表。原始数据表中的数据被定期处理，生成肯定对网络用户具有普遍意义的统计数据。生成这种统计数据所需的数据由统计处理模块按照预定的间隔时间从原始数据表中提取的。统计处理模块依次地生成所需统计数据，并使该统计数据存储到中央储存库的统计表中。

在优选实施例中，统计数据可包括应用程序在预定服务期间的最大、最小和平均响应时间，以及该应用程序在这个期间可用的时间百分比。

在本发明的另一个有关方面中，提供了在浏览者的计算机上交互式地顺序显示一系列数据集的装置。该数据集序列最好是图形，能显示某应用程序在一个可变周期，例如一个月中的可用性和响应时间，这种数据来自网络中一个或多个基于客户机的探测器。该数据集包含的数据元素对应的例如是，在所显示数据集中显示的月份中的一天，一组位于基于客户机的探测器的关于某应用程序的性能。数据元素动态地链接到其它数据集，从第一个数据集开始，数据集都将其自己的数据元素集关联到动态链接的数据元素，这些数据元素每一个都可进一步动态链接到其它这种数据集。动态链接最好能链接到第二个数据集中的数据元素，并从第二个数据集中的数据元素反向链接到第一个数据集中的数据元素。

动态链接的元素能响应浏览者的交互作用，诸如使浏览者的计算机从储存库请求相关数据集的鼠标点击或语音命令。这种请求引起一个基于软件的桥单元从储存库检索所请求数据，将数据集提供给浏览者的计算机，具有关联数据元素的相关数据集在该计算机上依次显示。

在本发明的一个实施例中，通过浏览者与图形显示的交互作用可以将图形显示转换成图表显示。

在另一个实施例中，第一个显示的数据集的数据元素表示一个或多个动态链接的相关数据集中每个关联数据元素的一个合并。例如，在本发明的一个优选实施例中，第一个数据集（该数据集例如代表，在指定月份的特定某日对在不同服务器上运行的某个应用程序监控的许多不同探测器的响应时间）的某个数据元素，可以动态链接到第二个数据集—第二个数据集所含数据元素代表在该指定月份的该特定日对在不同服务器上运行的该应用程序监控的探测器集合中每一个的响应时间。此外，第二个数据集中的数据元素可以动态链接到其中含有进一步相关数据元素的其它数据集。

这样，浏览者用启动动态链接的方法，就可以通过范围较宽的、关于一段时间内许多服务器上的应用程序性能的监控数据，“下探”到范围非常具体的、特定时间（某特定服务器上）某特定客户机处某特定应用程序的性能，并能再次从这个特定视图“上探”到相同的范围或更宽的不同视图。

本发明的主题，在本说明书的结尾予以特别指出并明确地对其提出权利要求。本发明的上述及其它特点和优点，在以下结合附图的详细说明中是显而易见的。附图简介：

图 1 表示本发明可以在其中实现的一个典型的网络计算机环境；

图 2 表示本发明一种采用单一探测器的简单实现方案，它能在客户机计算机监控服务器计算机上 Lotus Notes 应用程序的性能，并能记录和存储所监控性能数据供通过图形用户界面前端来显示，以及如果预定的性能阈值受到违反，则通过报警来提示支援人员。

图 3 更具体地表示了单一探测器实现方案所包含的操作序列，包括为探测器配置信息配备接口，探测器代码与服务器计算机上应用程序之间的交互作用，事务记录的生成与存储，以及报警信号的生成。

图 4 中的流程图表示循环进行的、客户机计算机上 AMA 探测器与服务器计算机上应用程序之间的服务请求和服务响应的交互作用以及事务记录的生成与存储。

图 5 表示将探测器生成的事务记录存储到中央储存库，以及对事务记录进行预定的统计处理，得出统计数据，装入中央储存库中的统计表。

图 6 的框图表示的 web 服务器计算机的功能是，按请求为图形用户界面提供连接浏览者计算机上 web 浏览器的小应用程序（applet），通过与浏览者计算机上图形界面的交互作用来完成浏览者生成的请求—该请求是通过用于访问数据库服务器计算机上中央储存库中存储的数据的软件桥单元进行处理的。

图 7 和 7a 表示浏览者计算机的前端图形用户界面的显示屏，显示的是对 1998 年 2 月 Poughkeepsie 位置所列全部服务器上运行的 Lotus Notes 应用程序的应用程序可用性和响应时间测量结果的可缩放测量。

图 8 以图表形式表示动态链接的图 7 和 7a 中所示的数据。

图 9 表示 1998 年 2 月运行 Lotus Notes 应用程序的 Poughkeepsie 位置上每个服务器的应用程序的可用性和响应时间数据。

图 10 和 10a 是呈现在浏览者计算机上的、由浏览者通过执行图 9 中对运行 Lotus Notes 应用程序的服务器 D01ML010 而启动的、关于 1998 年 2 月的可用性和响应时间的两种版本的缩放图形显示。

图 11 和 11a 是呈现在浏览者计算机上的、由浏览者通过执行图 10 和 10a 中对运行 Lotus Notes 应用程序的服务器 D01ML010 而启动的、关于 1998 年 2 月的可用性和响应时间的两种版本的缩放图形显示。

图 12 以图表形式表示动态链接的图 11 和 11a 中所示的数据。

图 13 呈示为响应浏览者执行的为启动以上在图 7～12 显示的动态链接的表示而采取的步骤的流程图。

图 14 显示的反向树图，表示由图 7～12 代表的动态链接显示的相互关系。

图 15 更加具体地展示了本发明的报警功能的特点。

以下说明要讨论的本发明的一个优选实施例，提供了一种监控、报告并根据服务器计算机上运行的 Lotus Notes（Notes 是 Lotus Development 公司的商标）应用程序的响应时间和会话可用性提供报警的系统方法和程序产品，其中从服务器计算机向通过分布式计算网络连接服务器计算机系统的客户机计算机系统提供应用程序的服务。尽管以下说明的性质是特定的，但本领域的熟练人员显然会明白，本文描述的发明性监控和报警技术，可以用于从分布式计算环境

中一个请求并接受另一个远程资源（即服务器）所提供应用程序的服务的网络资源（即服务器的客户机）的角度，对关于分布式计算环境上运行的任何应用程序的任何需要的性能特征进行评估。举例来说（不限于此例），本文的发明性系统、方法和程序产品就可以现成地采纳到公司内部网或外部网（extranet），用于对服务器计算机上运行并由客户机计算机通过内部网或因特网上的超文本传输协议（HTTP）访问的基于超文本标记语言（HTML）应用程序的可用性及响应时间进行评估。

为了更好地理解本发明的优点，首先考察一个能实现本发明的典型分布式计算环境，将具有指导意义。所以，图 1 表示了一种具有中小型企业计算环境的特点的典型分布式计算环境 100，这种环境能被扩展成包括更大的网络，诸如大型商务企业的公司网或内部网或者因特网。这种分布式计算环境 100 中可能会有一个或多个大小不同的网络 101、102，每个网络都有一组服务器计算机 104a、104b，它们能向同样连接到网络 103 的一组客户机计算机（其中一种典型客户机计算机可按照本发明执行监控程序）提供服务和信息。这种较小型网络 101、102 代表着例如企业的区域性机构所用的局域网，它们进而又能连接到一个或多个更大型网络 103。更大型网络 103 一般会连接这种较小型网络 101、102 以及其它服务器计算机 104c，相应地就使得例如这种较小型区域性网络 101、102 上的任何客户机或服务器计算机能够连接到其它小型局域网 101、102 或较大型网络 103 上的任何客户机或服务器计算机。

在下文的讨论中，应当明白所提及的服务器计算机对应的是这样一类计算机，它们是经过改造（经过编程设置）的典型的独立计算机系统，改造的主要目的是向客户机计算机的单个网络用户提供服务。客户机也可以是经过改造、能在网络上与服务器计算机系统（包括但不限于网络计算机 NC）交互作用的独立计算机系统或任何其它类型的数据处理系统。客户机计算机系统一般是指适于个人使用而不是其它计算机系统使用的任何计算机系统。

进一步应当明白，在本文中，图 1 所示的示范性分布式计算结构或网络 100，仅仅是有利地实现本发明的一种典型的分布式计算环境示意图。包括由这种分布式计算环境扩展到的因特网这种范围广泛的

分布式计算环境在内，这种相同的基本计算环境几乎有无限多的变型都能提供实施本发明的适当平台。

现在来更详细地考察本发明。参见图 2，图中表示的简化的计算机网络 200，包括一个按照本发明原则设计的应用程序监控与报警（AMA）探测器 201。浏览一下图中的网络 200 就可看出，该 AMA 探测器 201 是在与服务器计算机 202 相连的一个客户机计算机（例如图 1 中的计算机系统 106）中实现的。服务器计算机 202 包括应用程序 203，其性能是通过 AMA 探测器 201 进行的监控和报警活动而得到评估的。

操作中，AMA 探测器 201 通过请求服务器计算机 202 上运行的应用程序 203 的服务，建立一个与服务器计算机 202 的会话。会话的建立是由 AMA 探测器 201 通过网络连接 206 发给服务器计算机 202 的服务请求 210 启动的。相应地，服务器计算机的应用程序 203 将服务响应 211 通过网络连接 206 反馈给发出请求的 AMA 探测器 201。

特别值得注意的是，上述事务处理的次序，与计算机网络 100 的客户在其客户机计算机 106 上寻求获得服务器计算机 202 的应用程序 203 的服务时进行的事务处理的次序是完全相同的。实际上，通过从 AMA 探测器 201 的角度根据上述次序来监控应用程序并提供报警，本发明能在客户机计算机 106 从使用分布式计算环境中应用程序 203 的客户的角度，获得关于应用程序 203 的性能的现实描述。通过用这种方式执行 AMA 探测器 201，就有可能根据“最终用户”对基于客户机－服务器的应用程序 203 的经验来收集实时信息。根据本发明的一个优选实施例，AMA 探测器 201 将这种应用程序的可用性和响应时间或其它所需情况，相对于预定的并通常在合同中规定的网络 100 的性能标准加以量化。因此，如果依据从这种探测器获得的结果，那么，要建立给定网络上给定应用程序的这种性能标准并相应地确定网络上的应用程序达不到所建立标准的实例，就是一件相对简单的事情。通过本发明的装置，网络管理者就能建立 SLA 目标并监控对 SLA 目标的遵守。

更详细地考察图 2，注意到 AMA 探测器 201 可以从服务器计算机 202 接收许多不同的服务响应。例如，如果服务器计算机 202 的应用程序 202 恰当地响应服务请求，则 AMA 探测器 201 会从服务器计算机

202 收到一个成功地完成的请求的表示，即成功的服务响应。其它可能是，如果服务器计算机 202 不能用于响应服务请求 210，则请求过了预定的时段就会超时，AMA 探测器 201 就会根据在超时时限内没有收到返回的响应这个情况，对服务器计算机不可用作出记录。这可看作是一个不成功的服务响应 211。最后，如果服务器计算机 202 拒绝服务请求 210，则 AMA 探测器会再次将这个事务处理记录为不成功服务响应 211。被拒绝的服务请求 210 可能对应于各种不同的情形，诸如，客户机没有访问该特定服务器或服务器上应用程序 203 的授权，或者，应用程序 203 因为要进行维护而暂时“脱机”，或者，应用程序 203 由于种种原因而功能失常。

来自服务器计算机 202 上应用程序 203 的服务响应 211（包括无响应超时的确定），无论是成功还是不成功的，都在 AMA 探测器 201 被接收，探测器然后将事务处理的结果记录到数据库储存库 204 中。

数据库储存库 204 对于客户机计算机 106 上的 AMA 探测器来说可以是本地的，也可以是远程的，位于网络 200 上通过网络连接 207 连接到网络的另一点，或者，在优选实施例中，本发明也可以包括一个本地的和远程的数据储存库 204，数据可以在本地存储，并在同时或随后发送到一个中央远程储存库，后者能从许多在分布式计算网络 100 上不同点的不同应用程序进行监控的探测器收集探测数据。在优选实施例中，用于从网络 100 上多个探测器 201 记录事务处理记录的中央数据库储存库 204，被设计成能被分布式计算网络 100 的任何用户访问，理想情况下应提供一个前端图形用户界面（GUI）212，诸如基于内部网的 web 站点，该站点可通过超文本传输协议（HTTP）访问、并提供一页或多页超文本代码（即超文本标记语言或 HTML 页）以允许企业中任何人能容易地访问和分析其中储存的数据。本文随后将进一步详细叙述这种包括有基于图形的报告接口的中央报告机构。

在本发明的另一个优选实施例中，提供了适应对被监控应用程序 203 建立一组性能标准的装置。在这个优选实施例中，AMA 探测器 201 要判断，包括来自探测器的服务请求 210 和对应的来自服务器计算机 202 上应用程序 203 的服务响应 211 在内的完成的事务处理中，是否有任何有关成分超过了这种预定的标准。预定的性能标准可以包括的

尺度诸如有最大允许响应时间，和/或连续试图访问应用程序的服务的最大失败次数（一种会话可用性的指示）。

如果判定这些预定的性能标准中有的受到违反，就会促使 AMA 探测器 201 生成报警信号 208，送往报警机构 205，后者被设计用于向适当的支持实体通知这种违反，使得能够迅速地执行问题确定和补救步骤。

为了更好地阐示上述的事务处理序列，我们将参考本发明的一个优选实施例，其中，服务器计算机 202 上含有一个 Lotus Notes 应用程序 203。含有 AMA 探测器 201 的客户机计算机 106 进一步包含实现该探测器的功能的配置装置。

现在参见图 3 可以发现，这些配置装置包括一个基于 GUI 的前端 301，它用来从负责 AMA 探测器实现的一方（即从最终用户）引出关于探测器的功能的数据（即探测器配置数据 302）。在示范性实施例中，探测器配置数据 302 包括含有需要监控的应用程序 203 的服务器计算机 202 的名称（即目标服务器名）、目标服务器的网络地址和目标服务器上待监控的应用程序的类型（例如 Lotus Notes）。这个基本数据集对于建立网络通讯操作的初始序列是必要的，其中初始序列包含上述由 AMA 探测器 201 进行的事务处理序列—包括从客户机计算机 106 上的 AMA 探测器 201 向服务器计算机 202 上的应用程序 203 生成一个服务请求。

在本发明一个实施例中，可以引出的探测器配置数据 302 的其它数据项包括访问安全事务时可能需要的访问控制与验证数据（即诸如用户标识和口令之类的数据），以及关于对根据应用程序 203 的服务响应 211 收集的数据进行处理的指令。

对来自服务器计算机 202 上应用程序 203 的服务响应 211 进行处理的数据包括（但不限于）一个指示用于存储关于已完成事务处理的数据（即事务记录）的储存库 204 的存储器标志。正如前文所述，储存库可以是客户机计算机 106 的本地储存库 305 和/或远程的中央储存库 204，后者存储从对多个服务器计算机 202 上多个应用程序 203 进行监控的多个探测器 201 收集到的数据。在本发明的一个优选实施例中，存储操作既可以在包含 AMA 探测器 201 的客户机计算机 106 的

本地存储器 305 进行，又可以远程地在网络 100 上中央储存库 306 进行，代表两个储存库位置 305、306 的储存库标志都要提供。

在本发明的另一个实施例中，要引出探测器配置数据 302 来确定向服务器计算机 203 上应用程序 203 生成的服务请求 210 的采样频率。进一步将知道，可以指示 AMA 探测器 201 对在相同或不同服务器计算机 202 上运行的多个应用程序 203 进行监控。在这种多监控的实现中，可以局部地对每个要监控的应用程序定义采样频率，也可以全局地对客户机计算机 106 上由探测器 201 监控的全部这种应用程序 203 中的全部或部分定义采样频率。

如前文所述，本发明的一个实施例包含用于生成报警信号 208 的装置，报警信号用于启动报警机构 205 发出表示预定性能标准受到违反的信号。因此，这种实施例就要求建立这些预定性能标准，作为探测器配置数据 302 的一部分。例如，要为从被监控应用程序 203 接收成功服务响应定义一个最大可允许响应时间，是一件简单的事情。随后将要说明，AMA 探测器 201 生成一个事务记录，事务记录中包括自服务请求至服务响应的事务处理周期时间。如果这个时间超过预先设定的阈值，AMA 探测器就会生成一个报警信号 208 并发送给报警机构 205。

在本发明的一个实施例中，需要为应用程序 203 设定一个可用性阈值，作为探测器配置数据 302 的一部分，当阈值被超出时，会引发生成报警信号 208，实现这个过程的手段是，确定 AMA 探测器 201 从服务器计算机 202 上的应用程序 203 接收的连续不成功服务响应的最大数目。

用于为应用程序 203 确定可用性性能标准的 AMA 探测器 201 的某种特定配置，可能会招致—例如—按照前文说明建立一个可用性阈值，与配置信息结合，指示探测器 201 一旦接收到不成功服务响应 211 就立即重新启动服务请求 210，直到要么收到成功服务响应 211，要么阈值被超出，生成报警信号 208。这样，就有可能将基于应用程序可用性的报警信号 208 的生成，与某特定应用程序 203 的可用性的丧失紧密联系起来，这当然要依所定义的生成初始服务请求 210 的采样时间间隔而定。

分布式计算机网络 100 上的任何服务器计算机 202 都需要定期维护，维护时就要求服务器计算机 202 不得用于向网络上的客户机计算机 106 提供应用程序服务，所以最好考虑到这种“服务中断间歇”，以便在这些间歇时间不进行应用程序监控，或者调整在维护中断期间完成的监控结果，方法是将这些结果与某个标志或其它表示这些结果是在这种中断间歇期间获得的指示相关联。此外，索性还可以是这样的情况，即应用程序 203 不连续运行，只是在一定的预定时间才运行。因此，本发明的另一个实施例引出诸如应用程序的可用性时间表的信息，作为探测器配置数据 302 的一部分。

上述这些项每个都包含在本发明一个优选实施例的探测器配置数据 302 中。配置信息由图 3 中显示的基于 GUI 的 AMA 探测器界面 301 收集，图 3 进一步以表格形式表示了配置信息。按照图 3 中的图表实例，由界面 301 收集的探测器配置数据 302 被提供给客户机计算机上 AMA 探测器 303 的可执行部分，AMA 探测器接着按照 GUI 301 提供的配置信息 302，生成对目标服务器 202 上被标识的应用程序 203 的服务请求 210。

再次考察一次我们的典型实施例，其中，服务器计算机 202 上的 Lotus Notes 应用程序 203 要由 AMA 探测器 201 来监控。一旦探测器配置数据 302 已经被送入图形界面 301，驻留在客户机计算机 106 上的 AMA 探测器可执行代码 303 就采用用于服务器计算机上应用程序的应用程序接口（API）304，方式与最终用户客户机在网络 200 上寻求应用程序 203 的服务一样。然而，探测器代码 303 用来从应用程序接口 304 请求服务响应的访问频率、用户标识符、口令等，是由通过图形界面 301 提供的探测器配置数据 302 指定的。在 Lotus Notes 应用程序 203 的情况中，AMA 代码 303 采用 Lotus VIM（售主无关的报文发送）API 工具箱中的 Notes APIs 304。

在本发明一个包含基于 Lotus Notes 的监控器的解释性实施例中，探测器 201 和应用程序 203 之间的事务导致从探测器向 Lotus Notes 数据库服务器发出一个请求，要求打开一个名为 README.NSF 的 Lotus Notes 数据库文件。这个事务尽管不是通常的要求 Lotus Notes 数据库、邮件或中枢服务器（hub server）的用户请求，却一般指出了应用程序的可用性和响应时间。这样一个简单的服务请求

210 非常适用于本发明的监控目的，因为不要求通过口令验证请求会话来对一个成功的响应加密。此外，该特定服务请求 210 不需要在服务器计算机 202 上有处理器密集的操作，因此在计算机网络 200 上重复进行的该事务处理，实际上并不影响服务器计算机 202 为网络 100 上其它客户机计算机的总体性能。当然可以明白，对要监控的特定事务的选择，要留给管理探测器的实体，无论如何不应局限于本文中的特定说明。

这种客户机－服务器事务处理的完成包含服务请求 210 和服务响应 211，结果是生成一个通过 AMA 探测器代码 303 传递给存储器储存库 305、306 的事务记录 311。事务记录 311 包括有关该事务处理的信息，诸如应用程序 203 是否成功地响应了服务请求 210，从请求到响应的事务处理的整个延续时间，服务请求的日期，以及对于寻求监控特定应用程序的实体来说感兴趣的其它度量。

为了确定周期持续时间，AMA 探测器代码 303 中包括了有关定时器机构 307。在本发明的一个优选实施例中，定时器机构 307 只不过是这样一个简单机构，它从探测器 201 在初始服务请求 210 上放置一个时间标记，在探测器 201 的服务响应 211 上放置另一个时间标记，将这两个时间标记之差记录到事务记录 311。在含有 Lotus Notes 监控器的优选实施例中，可用性和响应时间是存储的事务记录 311 中包含的数据的关键度量。下面表示 AMA 探测器 201 的一个典型性的事务记录：

测量日期	监控器本地时间	目标服务器	目标服务器类型	服务器位置
"05/28/1997"	"19:54:40"	"002DBE01"	"SPM"	"SBY"

监控器名	0=F 1=S	IP Ping 时间	Notes DB 响应时间	请求间隔
"SBY"	"1"	"0.000000"	"1.281000"	"6"

由上面所列事务记录可以看出，该特定事务记录了一个成功的响应（其中域“0=F 1=S”中记录的是“1”），对这个事务的响应时间是 1.281000 秒。

除了向储存库 304、305 提供事务记录外，本发明的另一个特点是将事务处理的实时结果返回给基于 GUI 的界面 301 的有关域。这样，用户如果需要，就能在 AMA 探测器代码 303 处理了一个事务后，

通过浏览基于 GUI 的域 301，浏览对应该事务的响应时间和可用性信息。

正如上文讨论的那样，来自事务处理的记录可以本地存储在客户机计算机的储存库 305 中，和/或远程地存储在中央储存库 306 中。用于这种存储操作的指令包含在在图形界面 301 层指定的探测器配置数据 302 中。在一个替换实施例中，记录不存储在任何储存库中都是有可能的。例如会有这样的情况，即只希望从探测器对性能标准的违反报警，或者，网络管理者只对实时事务处理信息感兴趣。这种实现选择是应用程序监控项目的期望目标的功能，本发明都能实现。

再次参见图 3 会进一步明白，当生成事务记录时，探测器 201 进一步进行为了判断是否有任何基于界面（301）的预先定义的性能标准遭到违反而需要的阈值比较操作。如果判定这些标准有一个或多个受到违反，就由探测器 303 生成一个报警信号 308，传递给报警机构 205 去通知服务人员发生了违反事件。

图 4 总结了按照本发明处理一个客户机－服务器事务 400 所涉及的步骤。该过程由步骤 401 启动，行进到步骤 402，包括 AMA 探测器软件 201 的客户机计算机系统通过 GUI 模板或其它方式提示网络用户输入控制探测器代码 303 的功能的探测器配置数据 302。下一步在步骤 403，探测器配置数据被提供给 AMA 探测器代码的可执行部分，后者用该信息在步骤 404 向目标服务器计算机系统 202 上的被监控应用程序 203 发出一系列服务请求 210。如上所述，目标服务器 202 和应用程序 203 包含在步骤 402 中通过模板 301 提供给 AMA 探测器代码 303 的数据中。

由探测器生成的服务请求 210 引出的服务响应 211，在步骤 405 被探测器代码 303 接收。下一步在判断框 406 中，代码 303 判断服务响应是否是成功的。如上所述，如果是成功的服务响应，探测器接收一个表示服务请求已经由应用程序接收并且正在被完成的标志。相反，不成功的服务响应对应的要么是在预定的超时期限超过之前没有来自应用程序的服务响应，或者来自应用程序的服务响应指出该服务请求没有被履行。如果探测器 303 接收一个不成功的服务响应，探测器的可能回应（根据通过探测器配置数据 302 对探测器的设定）是，立即重试服务请求 404，如果能重试，则重试操作包含单一的或多个

重试企图，其次数受到重试次数或重试时间期限的限制，如判断点 407 指出的那样。如果超过了判定的重试延续间歇还没有引出成功的响应，记录一个不成功的服务响应的事务处理记录 410。

在任一情况下，不管该服务请求产生一个成功的或不成功的服务响应，都可以判断是否已违反任何预定义的阈值，这些阈值已经在步骤 402 在探测器配置数据中定义。如果判断出已经违反了这种阈值，由探测器 409 生成一个报警信号，并且传递到报警机构 205。

与此同时，或者紧接着阈值判断 408 之后，在步骤 410 有 AMA 探测器代码 303 生成一个事务记录 311。如上所述，事务记录 311 包括有关刚刚结束的特定事务处理（即服务请求和响应周期）的信息。然后在步骤 411，事务记录被存储到本地和/或中央远程储存库 305 和/或 306。在步骤 412，重复这个以步骤 404 发出连续服务请求为开始的周期，重复的条件是探测器配置数据有要求重复的指示，并且重复的频率也由探测器配置数据规定。最后，整个过程可看作是在结束框 413 处完成。

从以上说明可见，AMA 探测器 201 可以这样执行，即生成事务记录 311，重复地将事务记录附加到储存库中，储存库可以是本地的 305，也可以是中央的 306。这种中央储存库 306 含有在分布式计算系统 100 上每个被探测的服务器计算机 202 上的每个被监控的应用程序 203 的应用程序服务事务记录 311 的历史。本发明的另一个特点涉及向分布式计算环境 100 中的用户提供这种被存储数据的方法、系统和程序产品。本发明的另一个特点涉及处理由 AMA 探测器代码 303 生成的报警信号 308 并向支援人员报告报警情况的方法、系统和程序产品。

现在参见图 5，该图是表示的是中央数据储存库 504 的执行过程和组织结构的概览图，该储存库用于存储并为应用程序访问服务事务记录 311。更具体来说，图 5 表示的是将实时事务记录数据 311 从探测器代码 303 向中央数据储存库 504 的插入，以及根据插入的实时数据 505 对统计数据 506 的生成。

在操作中，当事务处理周期结束时由 AMA 探测器代码 303 生成一个事务记录 311。该记录被提供给一个数据库加载器模块 502，后者将该实时事务记录数据插入位于中央 AMA 数据储存库 504 中的原始数

据表 505 中。从数据库加载器 502 的记录插入的启动时机是在接收到来自分布式网络 100 中客户机计算机 106 上运行的任何探测器 201 的事务记录 311 时。由此可见，储存库 504 中的原始数据表 505 包含的实时数据的提供时间大约是事务记录 311 被探测器代码 303 生成的时间加上网络 100 上传递这种数据固有的延迟时间。

AMA 探测器 201 进行的应用程序监控活动的结果是，存在着建立在原始数据 505 基础上的可标识的统计数据集，它们是网络 100 的最终用户所感兴趣的对象。这些统计数据集例如包括，在确定的时间间隔内，成功的服务响应 211 占试图的服务请求 210 总数的百分比。这种预先确定的时间间隔一般和网络管理者与网络客户之间的 SLA 相一致（例如这种时间间隔一般对应于由 SLA 定义的 prime-shift 工作日）。另外，统计处理可以在原始数据表 505 中接收到预定数量的事务记录 311 时启动，例如当该表接收到 250 个事务记录时，启动对这 250 个事务记录的统计处理。当然，这个数值型阈值可以进一步细分成接收的相应于特定服务器 202 上特定应用程序 203 的监控的记录的数目。

通过以具有一定意义的间隔来处理中央储存库 504 内的原始数据，就能随时将有用的统计数据集 506 提供给希望确定网络 100 上应用程序 203 的某些性能指标的最终用户。从寻求关于被监控应用程序的性能的报告的最终用户有利的角度来看，表 505 中的原始数据被定期地预处理成统计数据集 506。由于实时统计计算是一项处理器密集的任务，所以这种对统计数据的预处理向网络中查询这种数据的最终用户提供了一种高效的报告机制。

统计数据的预处理是由于数据库报告器 507 的功能而得以实现的。报告器以预定的时间间隔访问由数据库加载器 502 插入到原始数据表 505 中的数据，由此处理数据，生成所定义的统计数据集，然后将其插入统计表 506 供最终用户查询访问。

数据库报告器 507 启动的间隔时间，与为计算统计数据所需的原始数据集的建立时间相一致。例如，在本发明的一个优选实施例中，在每个工作日的主班（即上午 7:00 - 下午 6:30）结束时，根据在探测器 201 重复生成并由数据库加载器 502 插入中央储存库 504 的原始数据表 505 的原始数据，进行一次计算。另外，如前文所述，也可以

在原始数据表 505 中收到预定数量的事务记录 311 的基础上进行计算。

这种插入的原始数据对应于来自在此间隔时间内一直监控着应用程序 203 的 AMA 探测器代码 303 的事务记录 311。在这个间隔时间结束时，将新插入的原始数据提供给数据库报告器 507，后者接着预处理该数据以便提供关于在此间隔时间内各被探测的应用程序的可用性（即确定在该间隔时间内所记录的各应用程序的成功地服务响应的百分率），是一件简单的事情。此外，统计数据中还可以增加一系列为预处理间隔时间而记录的响应时间。预处理的统计数据然后被提供给中央储存库的统计表 506。这样，这种数据只计算一次就能很快提供给查询中央储存库的最终用户。插入统计表 506 中的统计类型，通过随后关于对图 7～14 所作说明中所述的提供给中央储存库 504 的浏览者的前端 GUI 的讨论，将更加明显。

在一个典型实施例中以及随后结合图 6 所述的那样，最终用户可以在 web 浏览器应用程序上接收这种被存储的数据，方法是连接到 web 服务器计算机 508 内的 web 服务器程序 600，该计算机一经要求就能访问在中央储存库 504 中存储的统计数据，无须对统计数据进行处理。

现在参见图 6，进一步阐释 web 服务器计算机 508 的功能。中央储存库 504 驻留在数据库服务器 601 上。寻求了解网络上应用程序性能数据的最终用户，通过计算机系统的 web 浏览器应用程序 602 建立一个与 web 服务器程序 600 的一个会话 608，计算机系统例如是（可能包括也可能不包括 AMA 探测器代码 201 的）客户机计算机系统 106，能够连接到数据库服务器 601 和 web 服务器计算机 508 中的 web 服务器程序 600。在优选实施例中，web 服务器程序 600 存储一个软件包 603，软件包可以是个 Java 类文件。软件包 603 按请求被提供给最终用户计算机 106。当在最终用户计算机 106 收到它时，它被 web 浏览器 602 解释，以便提供 AMA Java 小应用程序 604，后者的功能是向 WBE 浏览器 602 上的最终用户提供一个关于所需性能数据的图形标志（Java 是 Sun Microsystems 公司的商标）。当最终用户通过统一资源定位器（URL）与 web 服务器程序 600 通讯时，就根据 web 浏览器程序 602 提供的信息启动一个验证过程 608。一旦经过验证，就从 web

服务器程序 600 提供（608）HTML 代码给 web 浏览器 602，然后，将基于 Java 的软件包 603 提供给（608）web 浏览器 602。

在最终用户计算机收到的 Java 软件包 603 被 web 浏览器 602 解释，浏览器生成 AMA Java 小应用程序 604。AMA Java 小应用程序 604 起着最终用户和储存库 504 中储存的数据之间的接口的作用。AMA Java 小应用程序 604 包括一个 GUI（AMA GUI 605）。AMA GUI 605 包含一个图形和图表对象的集合，用于辅助最终用户请求数据（609）和分析随后从数据库服务器 601 上的数据库储存库 504 提取（609）的数据。作为 AMA 小应用程序 604 的解释的一部分，在最终用户计算机 602 和 web 服务器程序 600 之间建立的会话 608 结束。一旦结束，就不必利用 web 服务器 508 来进行为了访问储存库 504 中储存的数据所需的计算和通讯，于是 web 服务器 508 就能用于服务其它寻求与其建立会话的最终用户。

最终用户计算机 106 下一步通过 AMA GUI 605 生成一个对储存库 504 数据的请求，其效果是启动一个与数据库服务器 601 的临时数据链路，用于接收储存库 504 中储存的数据。AMA 桥 606 是一个基于软件的接口机构，逻辑上通过连续的通讯会话 610 与储存库 504 相连。设定桥 606 访问储存库 504 中用于由最终用户通过运行 AMA 小应用程序 604 的 web 浏览器访问的一部分。前文说过，最终用户的验证是在 web 浏览器 602 与 web 服务器程序 600 之间的会话建立期间完成的。所以，没有必要验证最终用户通过桥 606 对储存库 504 的访问。AMA 桥 606 通过 AMA 小应用程序 604，管理最终用户会话 602 与数据库服务器 601 上储存库 504 之间的高速双向数据链路 609。这样，用于通过 AMA GUI 605 用图形方式表示探测器监控结果的数据密集的图形表示材料被提供作为在最终用户的 web 浏览器 602 上本地实现的软件的一个包 603。与储存库 504 中存储的探测器监控结果对应的相对有限的数据在数据库服务器 601 和最终用户的计算机 106 之间传递（609）。这个数据被 AMA 小应用程序 604 处理，用于通过 AMA GUI 605 显示出来。

用户-数据库之间的临时数据链路 609 是由用户与用户 web 浏览器 602 上 AMA 小应用程序 604 中所含 AMA GUI 605 的交互作用而启动的。例如，用户通过点击在用户 web 浏览器 602 上本地显示的模板

605，使其计算机 602 请求在数据库服务器 601 上储存库 504 中存储的数据。AMA 桥软件 606 起着在最终用户的 web 浏览器 602 与数据库服务器 601 上储存库 504 之间通过数据链路 609 通讯的相对有限的数据的高速传输接口。桥软件 606 以守护程序的形式实现，逻辑上通过通讯会话 610 连续地连接到储存库 504，典型实施例中的桥软件物理上可以驻留在 web 服务器 508 上、数据库服务器 601 或一个中间计算机（图中未于示出）上。桥守护程序 606 为最终用户的 web 浏览器 602 上运行的小应用程序访问 AMA GUI 605 提供服务。桥 606 通过 GUI 605 接受最终用户的请求并用储存库 504 中的适当数据作为响应。

采用 Java 数据库连接（JDBC），桥守护程序 606 连续地连接到储存库 504，使得对储存库的多个请求不会要求创建多个数据库进程（而如果采用其它的访问形式，诸如通过公共网关接口(CGI)，则会出现这种情况）。相应地，任何数量的最终用户的多个请求通过桥守护程序用 FIFO 队列（或其它方法）加以管理。这样，桥守护程序 606 为将最终用户直接连接到中央储存库 504 而提供了一种高效的机制。

进一步将明白，多个最终用户（n）每个都可以在给定的时间访问 web 服务器程序 600，每个都可以以 Java 类文件被提供。相应地，由于最终用户与 web 服务器 508 之间的会话，一旦在提供了 Java 包 603 后就停止，在这期间，大量的最终用户（m，其中可能 m>>n）可以本地地执行包括 AMA GUI 605 在内的 AMA 小应用程序 604，可以建立到桥单元 606 的数据链路 609。在桥单元与储存库 504 之间建立的单一连续通讯会话 610 将通过在最终用户计算机与桥 606 之间建立的 m 个数据链路上发送被请求的数据，顺序地将这 m 个最终用户计算机 106 连接到在储存库 504 中存储的数据。

最终用户在建立与 web 服务器程序 600 的会话时，通过 web 服务器程序 600 中提供的一个主菜单模板，指出其所感兴趣的是哪组服务器上的哪个（些）被监控应用程序。web 服务器程序 600 接收这个信息，为在该最终用户的 web 浏览器 602 上创建包括 AMA GUI 605 的 AMA 小应用程序 604 提供（608）适当的 Java 包 603。最终用户与其计算机 106 上 web 浏览器 602 内本地执行的 AMA GUI 605 的交互作用，启动到桥守护程序 606 的数据链路。桥监护程序连续地连接到储存库 504，并管理着它们之间的数据交换。

从以上的说明中明显可见，数据库储存库504通常可以包括大量的事务记录集311、505和关于已经/正在被分布式计算网络100上的多个客户机计算机106上执行的多个AMA探测器201监控的多个被监控应用程序203的性能的统计数据506。进一步会明白，这种被存储数据的有用性直接关系到用来分析数据的手段。因此，本发明的另一个特点包含用于以简明的方式将所存储的应用程序监控数据504呈现给最终用户的图形显示装置。

具体来说，本发明提供的交互式报告生成和图形表现装置，使浏览者能查询并获取与分布式网络100上被监控的应用程序性能相符的图形和图表数据查询和获取数据可在两种层次上进行。在高层次上，例如能将运行一个应用程序或多个应用程序的多个服务器的可用性和响应时间，在一个月或更长时间内收集的性能数据的基础上显示出来；在粒度范围增加到一个显示或表的层次上，该画面或表中包含对应于运行着特定被监控应用程序的特定服务器每小时的性能的数据。每个表或图中的这些数据层每个都动态地连接到后继的表和图、连接到上一个表和图，所连接到的这些表和图的数据显示范围或者更宽，或者更窄，可供用户按需分析。这些连接的表和图的遍历提供了一种对用户友好的查询工具，用于从中央储存库504检索存储的数据，用于以便于用户理解的方式在不同的规定层次显示数据。

尽管事实上在以后的描述中将用计算机鼠标器操作来说明与表示系统的交互作用，应当明白，在本发明范围内，浏览者可以用任何已知的计算机界面机制与这些图形或图表表示进行交互作用，包括但不局限于这样一些指示设备，诸如计算机鼠标或轨迹球、游戏杆、触摸屏或光笔设备，或是与计算机系统的语音识别交互作用。用户的交互作用指定要向浏览者显示的数据的连续层次。

现在转至图7-12，图中展现的是一个典型的顺序的浏览者交互式图形和图表表示集，它是作为本发明优选实施例中AMA GUI 605的一部分实现的。

图7中的图700是在用户填充沿着图700底部的模板域707时生成的。域707包括要显示的AMA特征（在所示例子中是响应时间和可用性）、运行着要显示其监控结果的应用程序的服务器（或者如本例所示的所有服务器）以及监控的时段（本例中是1998年2月）。

用户通过填充这些域，使AMA GUI 605生成一个向桥守护程序606的请求，桥守护程序于是查询储存库504，将数据返回到GUI，用于在图700中显示。在我们接着描述下述图7-12中的动态连接显示的特点时将会明白，各个表或图形上的互连连接或活动区，当被用户启动以遍历随后描述的图形和图表系列时，每一个导致一个对要从AMA GUI 605向桥守护程序606传递的数据的请求，并进一步导致桥守护程序606查询储存库504并将所请求数据从储存库返回到GUI 605，后者接着通过AMA小应用程序604将返回的数据装配成诸如图7-12的图形和图表中所示的一个显示格式。

经过上述说明后，我们转向图7，该图表示的图形700显示的是1998年2月份在一组AIX服务器上的Lotus Notes邮件应用程序(Lotus Notes(AIX) Mail)的响应时间和可用性。图形700的X轴对应该月的日子。图形的Y轴上的第一种刻度701对应着被监控Notes应用程序该月份逐日的可用性百分比指示，第二种刻度对应着被监控应用程序事务处理的响应时间(秒数)指示。响应时间和可用性的范围可以按需放大或缩小，方法是点击分别为每种刻度702a和701a显示的“+”、“-”标志。例如，图7a表示的图形700a与700相同，其响应时间刻度702被扩展了，这是通过702a的“+”功能“放大”的。

再次参见图7，图形700包括与响应时间的测量对应的条703，与可用性的测量对应的数据点704。条表示703中包括彩色编码的中间点705，该点对应的是逐日监控的应用程序的响应时间的第50百分点。换言之，应用程序对服务请求210的响应211在所说该日的响应时间的量度有一个范围(例如从0.1秒至10秒)，中间点705代表的响应时间量度是，其余的每日量度有一半更高(更慢)，其余的每日量度另一半更低(更快)。在解释性例子中，条703以图形方式显示所说该日测量的除了最快的5%和最慢的5%以外的响应时间(即范围跨度在第5百分点到第95百分点的响应时间，或另由SLA定义)。点704中的每个单一数据点对应每一天，表示该日获得成功服务响应211的服务请求210的百分比。

例如，参见特定的图700，可以看到，2月2日这一天的响应时间的指示范围下限0.1秒，上限假定超过4.0秒，第50百分点落在0.1秒

左右。这表示，1998年2月2日的大多数响应时间都聚集在十分之一秒的区域内。同样，图700表示2月2日的应用程序可用性在98%左右。

按照以上说明将会注意到，图700中包括“热点”(例如704)或“活动区域”，它们是该图的交互作用部分。这些热点超连接或者交互式地连接到允许浏览者更详细地动态检查图700上特定兴趣点的有关图形和图表。例如，如果浏览者将其鼠标点置于该图的背景区域708(即不在显示特定某日的数据的区域，随后将要说明)后点击鼠标，则该浏览者就会启动一个对对应于图700中所示数据图表表示的数据的请求，相应地就会出现一个图8所示的图表800。所以应当明白，图700的背景708是一个交互式热点。

表800表示创建图700所用数据的图表形式，其用途在于能使人更精确地阅读对应特定某日801的数据。为了阐明这一点，我们来检查一下2月2日的数据。现在我们可以从表800看到，最快的响应时间802实际上是0.071秒，重要的是，最慢的响应时间803是14.062秒(图形表示700限制在4.0秒，但是可以采用上述如表示2月2日的响应时间范围在14秒的表700a中所示的缩放功能702a重新调整)。此外，按日的可用性数据804和按日的中间点响应时间805也以图表形式列举。用鼠标点击表800，能动态地使浏览者返回到表700。

如果某浏览者希望分辨在图700、700a和表800所示的慢速14.062秒响应时间是在哪个服务器、哪个时刻发生的，可将鼠标指针放在条703或代表该日的行801上，点击其鼠标，AMA GUI将动态地查询储存库504，检索有关数据，向浏览者展现图9所示的表900。

表900表示的是感兴趣的日子(即1998年2月2日)在图700上表示的每个服务器的可用性和响应时间统计数据。各服务器由位于表900左边部分的超连接的或以其它方式动态关联的按钮901(热点)来代表。表中各栏有代表该日的可用性百分率的栏902，以及该日的5%(903)、95%(904)和中间点(905)响应时间。

通过在表900中表示每个服务器的统计性能，现在就有可能容易地确定感兴趣的日子里的哪个或哪些服务器表现了较差的响应时间。这些服务器就是导致在图700上出现较差响应时间的图形表示的机器，它是表900中所示服务器的一个合成。当然可以明白，单独列

举的服务器中的每个服务器的95%响应时间，许多都不在图700上为综合的服务器显示的95%响应时间中显示出来。

应当明白，这种剔除仅仅是实施细节上的选择问题，然而，它被包括在优选实施例中，原因在于人们发现，将这些外落的数据点(即响应时间)包括进来，歪曲了所说期间大多数监控数据的图形表示。所以，通过剔除这些外落点，人们发现显示数据更精确地代表了被监控应用程序的典型性能。

查看表900可以看见，服务器D01ML010在2月2日的95%的响应的响应时间是5.649秒。点击对应该服务器的按钮901，就会出现如图10所示的图1000。

图1000以与图700所述的类似方式，显示了服务器D01ML010的性能(即可用性和响应时间)。图中描述了服务器在1998年2月每天1003的响应时间1002和可用性1001。

与图700中的方式类似，条1004代表被监控程序每天的响应时间。不过该图中只描述了服务器D01ML010的数据。如图700中一样，彩色的散列标记(hash-mark)1005对应当日5%－95%的被记录的响应的中间点。各数据点1006对应于各天应用程序的可用性的读数。我们知道，服务器D01ML010有95%的响应时间是5.649秒，所以该读数超过了图1000的上界，但是在图10a的图1000a中，响应时间的边界已经扩展到能显示该服务器在1998年2月2日的该5.649秒的响应时间。

假设浏览者想要检查以图表格式表示的该数据，可以在背景区点击鼠标(或通过其它这类基于浏览者的交互作用)，就会调出这种数据的图表表示，其形式与表800相同，只不过该表代表的是与服务器D01ML010对应的表1000中的数据。由于该表几乎等同于表800，所以本文没有必要将其展示出来，然而重要的是要注意，每个图形表示都动态连接到一个更具体地展示该图形表示的内在数据的表。

再次参见图10，假设浏览者希望检查服务器D01ML010在1998年2月2日的每小时的性能图形，他/她只要将指针放在对应该日期的条1004上，点击鼠标，就能导致动态生成图11中所示的图1100。

图1100的格式等同于图700和图1000，但是图1100是按小时1103描述服务器D01ML010在1998年2月2日的可用性1101和响应时间1102

的。采用相同的标记1104、1105和1106作为响应时间和可用性的标志，但是图1100上的条现在表示最大、中间点和最小值。

浏览者可以从这个图示1100中判定，服务器在该日的哪个（些）时间表现了较差的响应时间性能。相应地，通过检查2月2日服务器D01ML010的图1100我们可以看到，在上午10时、上午11时和下午3时，响应时间看起来格外慢。然而，由于图1100显示的响应时间的界限在4.0秒，所以浏览者要按下与响应时间尺度1102a关联的“+”按钮，才会出现图11a中的图1100a，从中可以看到，上午10时的响应时间超过27秒，上午11时的响应时间约为25秒，下午3时的响应时间约为6秒。

假若浏览者希望查看该图1100的细节，他/她只要点击图上的任何位置，就会出现如图12中所示的表1200，后者依次展示包含图1100的性能数据的图表表示。从该表中可以看到，运行着被监控的Lotus Notes（基于AIX的）邮件应用程序的D01ML010的响应时间，实际上在上午10点期间最坏的情况是73.281秒。

图13表示本发明提供的显示与报告生成装置的一个概述。总之，本发明提供一种用于与储存库504的数据的图表或图形显示表示交互作用的机制。在步骤1301，用户请求检索和显示第一个数据集（即在2月份服务纽约营业处所的Poughkeepsie的所有服务器的Lotus Notes（基于AIX的）邮件应用程序的可用性和响应时间数据(参见图7)）。在步骤1302，GUI 605将该请求传递给桥守护程序606，后者依次查询储存库504并将检索出的数据返回给GUI，GUI在步骤1303显示所请求的数据。在步骤1304，用户与该显示上的指定热点交互作用，结果依次启动一个对与在第一个显示上出现的并被交互作用的数据相关的第二个数据集的请求。在下一个步骤1305，GUI 605再次将该请求传送到AMA桥守护程序606，后者在储存库504中查询所请求的相关数据，该数据通过桥被提供给GUI 605（步骤1305），后者在步骤1306在从储存库返回的该数据的基础上构造对第二个相关数据集的显示。

在可与图13结合起来参阅的图14中可以看到，以上描述的连续显示的相互关系可以看成是一个倒置的树图1400。从框1401开始，浏览者通过其计算机系统上的菜单（或其它形式）请求显示一个数据集，

包括如框1402所示的应用程序监控数据。注意到框1402中既包括由条形图1402a示意性地代表的监控数据集的图形表示，也包括如表1402b所示的监控数据集的图表表示。该表和图由连接1402c动态地相互连接，用户只要点击（或以其它方式交互作用于）浏览者计算机系统上的活动显示区，就能在监控数据集的图表表示与监控数据集的图形表示之间来回切换。

在表1402b或图1402a所示数据集内的是以A和B代表的数据元素。表和图中的这些数据元素，每个都能被实现为前文所述的活动区或热点，使得它们能通过点击鼠标或用其它的浏览者界面技术被指定。根据浏览者的指定，图13中所示的动态连接活动就被执行，提供对包括关联数据元素的第二个数据集的显示。例如，假设浏览者要指定表1402b或图1402a内的数据元素A，就要启动图13中所示的步骤序列，用于动态连接1403a，随后显示由框1404代表的表和/或图形显示。

框1404中显示了被显示第二个数据集的图形表示1402a和图表表示1404b。第二个数据集包括数据元素A1、A2和A3，它们与图1402a或表1402b所示的数据集合中的数据元素A相关联。在优选实施例中，这些关联元素每一个代表数据元素A的一个部分。例如在一个其中数据元素A代表记录到的在给定日期某Lotus Notes应用程序对三个不同探测器（即探测器1、探测器2和探测器3）应答的响应时间的范围的实施例中，1404a和1404b中所示的数据集包括在该给定日期在各探测器记录到的响应时间的范围，使得A1代表探测器1在该给定日期的响应时间，A2代表探测器2在该给定日期的响应时间，A3代表探测器3在该给定日期的响应时间。当然应当明白，图1404a和表1404b是通过连接1404c互相动态连接的，可以按上述数据集1402a和1402b通过连接1402c切换的类似方式进行切换。进一步应当明白，数据集1402中的每个数据元素（即A和B）都可以通过1403a和1403b连接到其它的表示，其方式如对数据元素A描述的一样。

1404a和1404b中所示的数据集中的每个数据元素，可以通过连接1406，进一步动态连接到其它依次包括代表着数据集1402a和1402b中数据元素的部分（即数据元素A1、A2和A3）的数据元素的数据集（未予示出）。

从上述典型实施例中应当明白，AMA GUI与桥606和储存库504结合，能使最终用户迅速有效地分析大量的数据记录。在以上的简化例子中，允许浏览者从服务器性能的概括图开始“下探”，将响应时间的异常局部化到特定时刻的特定服务器上。

进一步应当明白，上述技术可以结合到对应特定服务器的性能的可用数据，诸如IBM的Netfinity或其它此类产品提供的数据，以进一步精炼问题确定的过程。这种确定过程的粒度仅仅受到为给定网络而收集的网络性能量度的数量和类型的限制，以及用于将这种收集的数据关联的数据筛选和分析工具的局限的限制。

本发明的另一个发明特征引入了用来在判定任何定义的性能标准受到违反是向服务人员提供报警的技术。

如上所述，AMA探测器201可配有探测器配置数据202，包括诸如最大响应时间或最小应用程序可用性的阈值数据。这些标准受到AMA探测器代码303的监控，当检测到违反标准时，探测器就生成一个报警信号308，用于向报警机构205通知这个违反。报警机构205接下来的作用是向支援人员发出表示发生违反的信号，或以其它方式启动一个问题解决响应。

图15更详细地展示了这种报警序列1500的典型实现。当AMA探测器代码303判定，定义为探测器配置数据202一部分的某阈值受到违反时，报警信号308就被发送到报警机构205，在优选实施例中，报警机构被表示为一台运行着IBM NetView（R）软件应用程序的服务器计算机1501（下文称为NetView服务器）。在优选实施例中，阈值的违反导致AMA探测器代码生成一个软件标志，称为陷井308。这个陷井标志是按照用于基于客户机－服务器网络的通讯的传输控制协议/互联网协议（TCP/IP）、简单网络报文传递协议（SNMP）配置而定义的。这个陷井标志包括关于服务器计算机202和其中违反标准的应用程序203以及记录到的违反类型的数据。上述软件陷井对于本领域的熟练人员来说都是熟知的，因此不再详细解释。进一步应当明白，尽管本发明采用软件陷井来发出表示发生违反的信号，其它机制诸如电子邮件或寻呼、弹出式屏幕或其它通知方法也是可以采用的。

陷井308一旦在NetView服务器1501被接收，就被附加到服务器内的储存库trapd.log 1502中。储存库1502被另一个服务器1503（图中

表示为IBM Global Services Coordinator(GSC)服务器）以预定的时间间隔依次扫描。

GSC服务器1503将trapd.log 1502中的任何新条目与GSC服务器内的一组表1504进行比较。这些表1504包括的数据的相关内容是，应用程序的类型、服务器位置、违反类型、要向其通知违反的服务人员、向服务人员通知的方式（即电子邮件、寻呼等等），通知方式基于的因素诸如有违反的严重程度和日期时间。

如果GSC服务器能够将trapd.log 1502中的服务器、应用程序类型和违反类型与其表1504中的一个条目匹配，就按照规定的方式（即寻呼机、电子邮件等）1506向指定方生成一个报警信号1505。

如果AMA探测器代码303随后判定，受违反的阈值已经返回到可接受的水平，就采取进一步措施来取消报警。在这种实施例中，取消的步骤与原来的违反指示的产生次序相同，只是原来要生成一个软件陷井308，而现在则要指出违反已经解决。陷井被存储在trapd.log 1502中，被GSC服务器扫描，GSC服务器的表1504会指出报警应当取消，于是要么在发送（1506）之前停止报警信号（1505），要么发送（1506）一个要求取消上一次报警的消息1505。

尽管本文详细描述了优选实施例，对有关领域的现在和将来的熟练人员来说，显然，可以进行各种修改、增加、改进和加强而不偏离本发明的精神，它们因此应视为在随后的权利要求中所定义的本发明的范围之内，应当按照权利要求来对首次披露的本发明保持适当的保护。

说 明 书 附 图

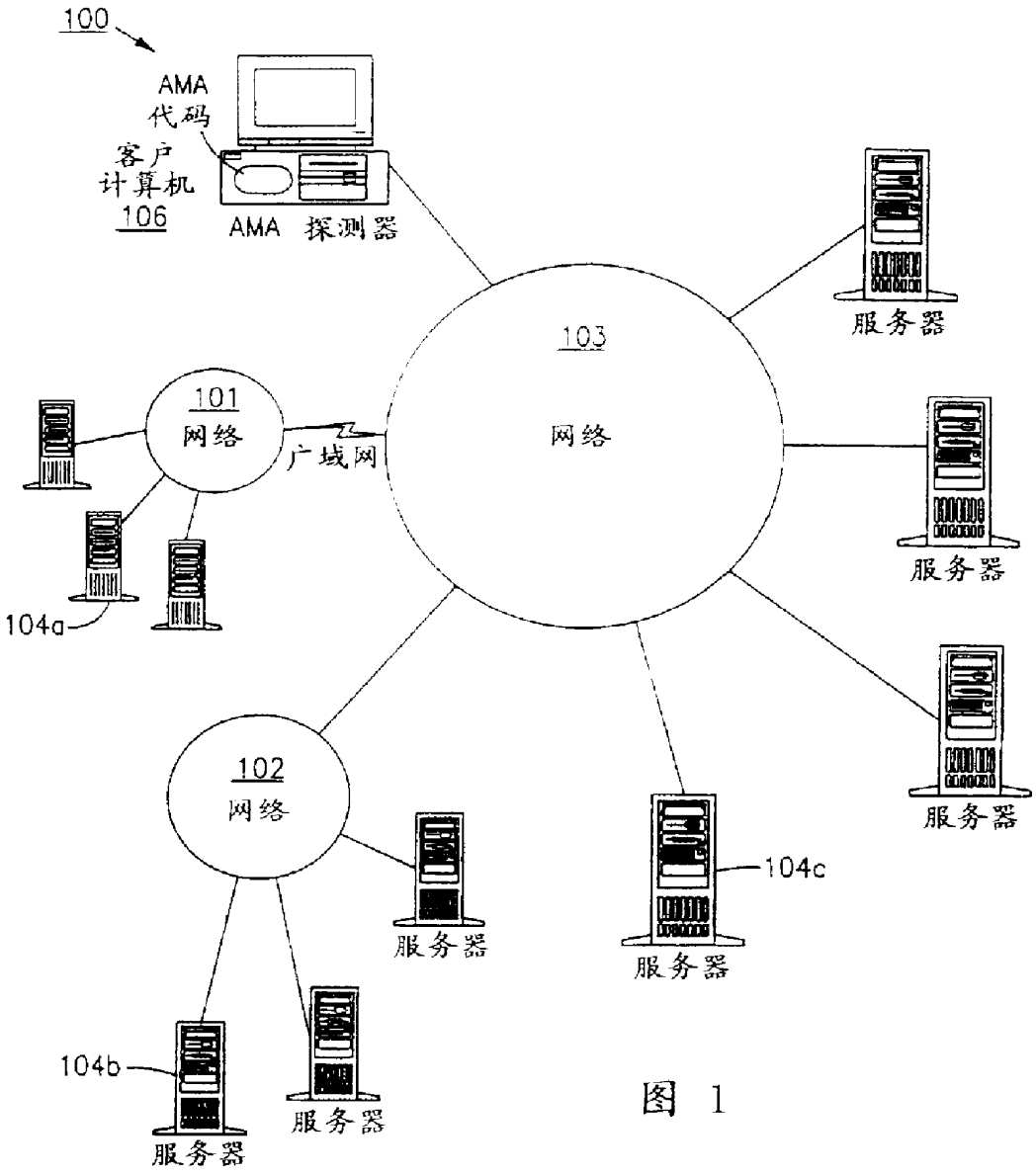
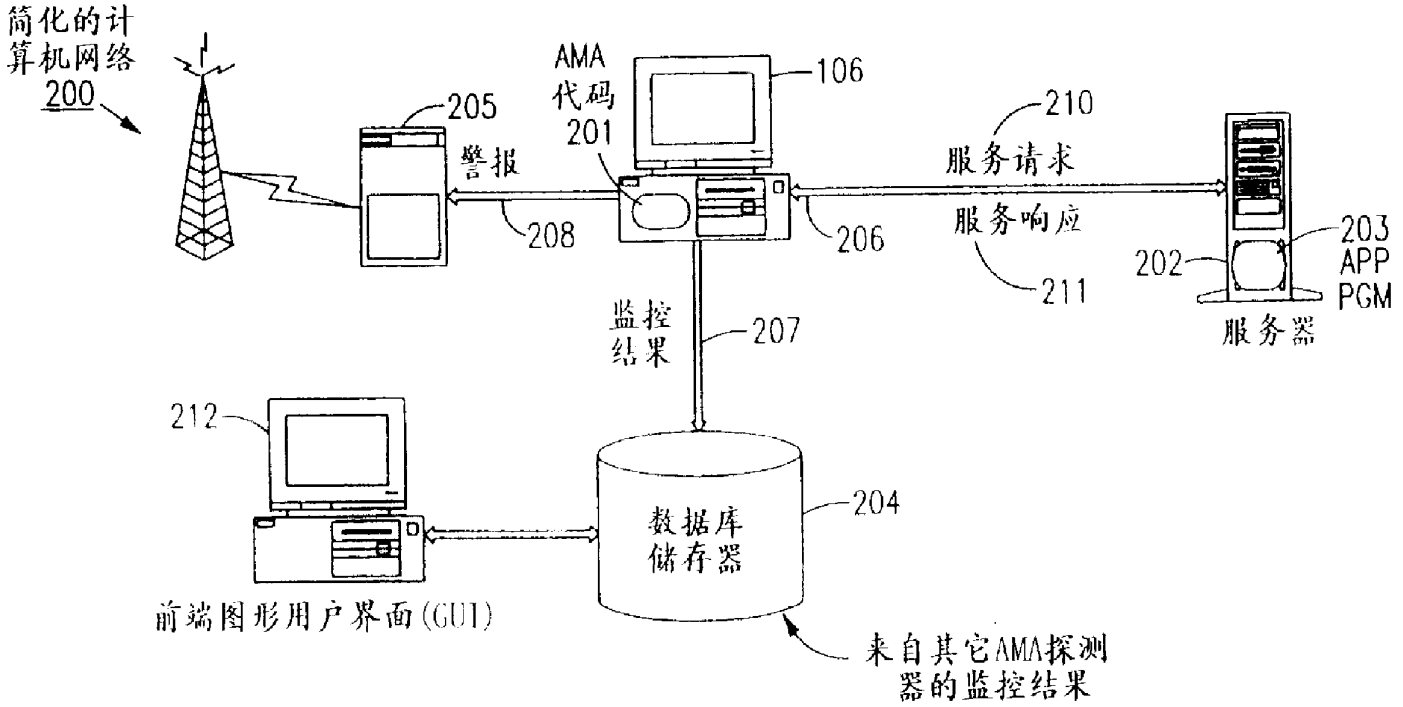

100
AMA
代码
客户
计算机
106
AMA 探测器
103
网络
服务器
101
网络
广域网
服务器
104a
102
网络
服务器
104c
服务器
服务器
104b
服务器

图 1



服务响应(211)
- 成功
- 不成功
  - 超时（服务器不响应）
  - 服务请求被拒绝

图 2

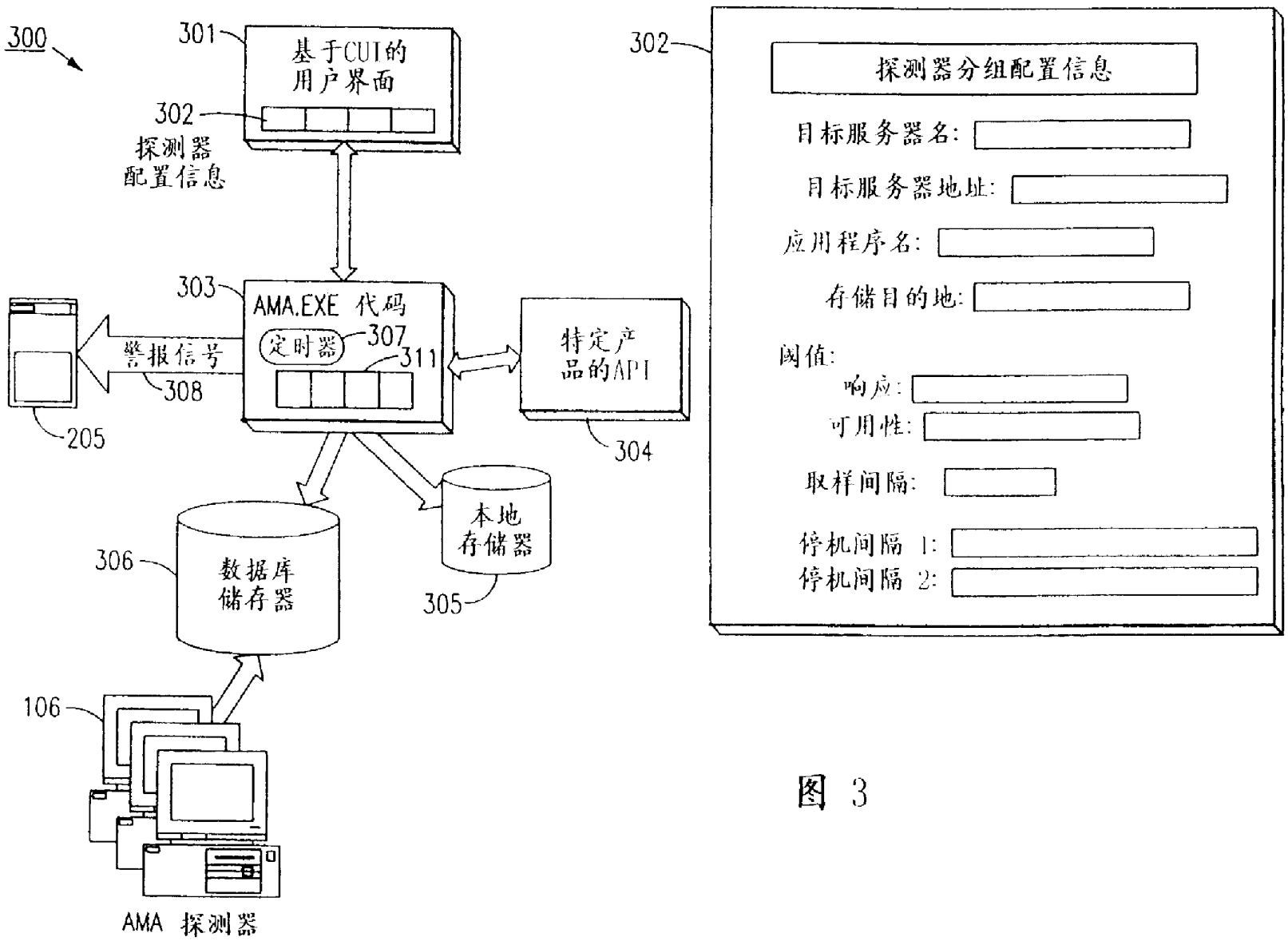

300
301
基于CUI的
用户界面
302
探测器
配置信息
303
AMA.EXE 代码
定时器
307
311
警报信号
308
205
特定产
品的API
304
本地
存储器
305
306
数据库
储存器
106
AMA 探测器
302
探测器分组配置信息
目标服务器名:
目标服务器地址:
应用程序名:
存储目的地:
阈值:
响应:
可用性:
取样间隔:
停机间隔 1:
停机间隔 2:

图 3

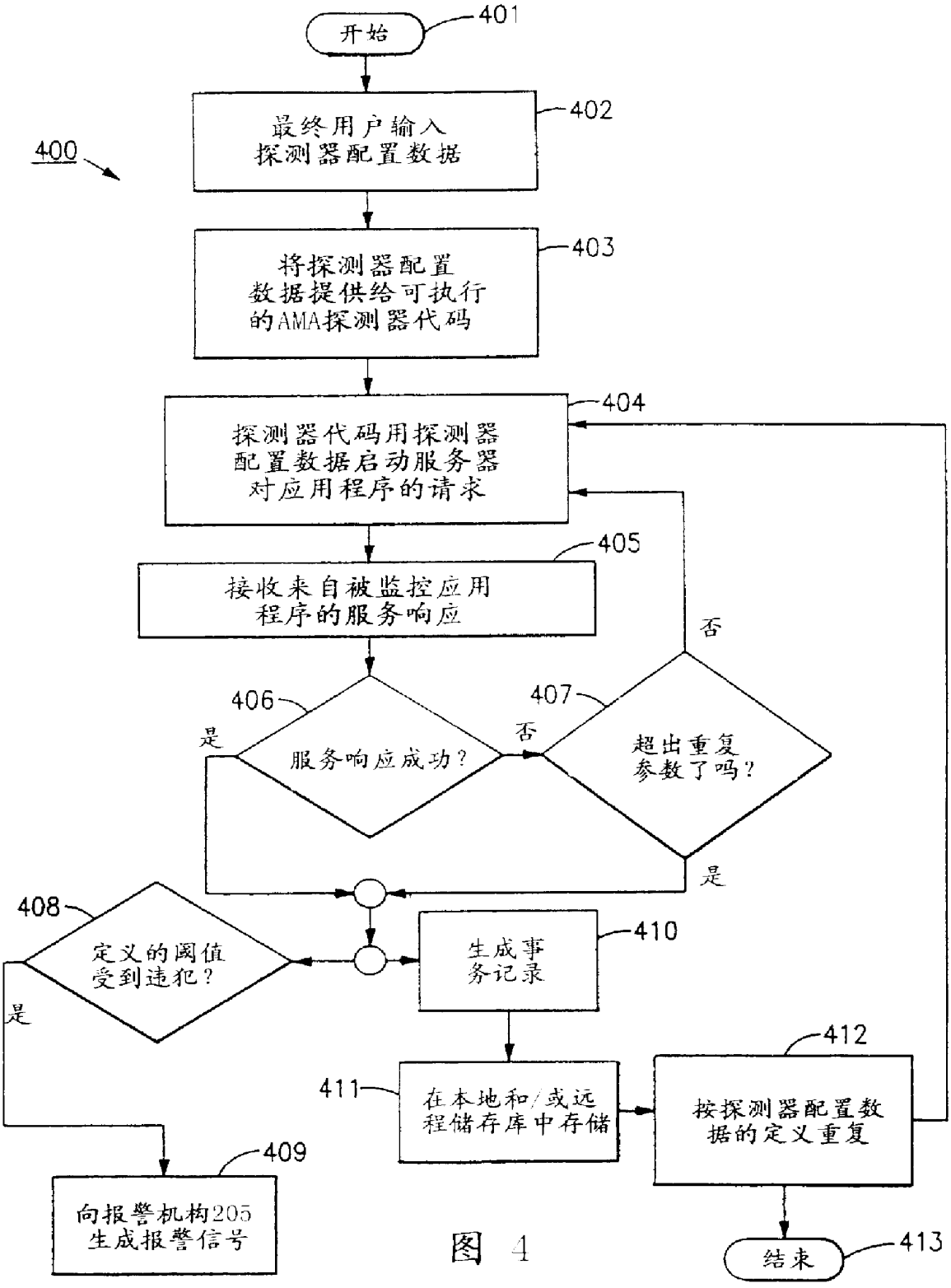

400
开始
401
最终用户输入
探测器配置数据
402
将探测器配置
数据提供给可执行
的AMA探测器代码
403
探测器代码用探测器
配置数据启动服务器
对应用程序的请求
404
接收来自被监控应用
程序的服务响应
405
406
服务响应成功？
是
否
407
超出重复
参数了吗？
否
是
408
定义的阈值
受到违犯？
是
生成事
务记录
410
411
在本地和/或远
程储存库中存储
412
按探测器配置数
据的定义重复
409
向报警机构205
生成报警信号
结束
413

图 4

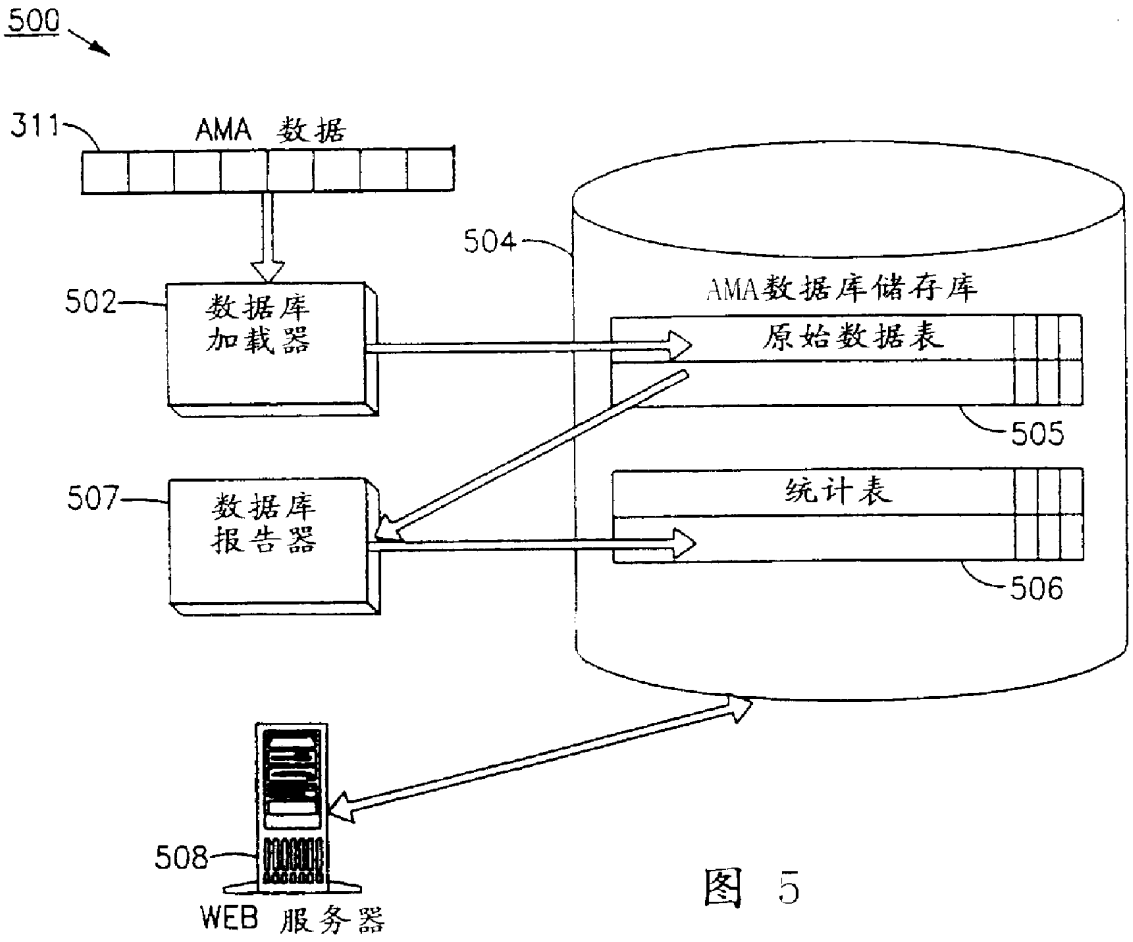

500
311
AMA 数据
504
AMA数据库储存库
502
数据库
加载器
原始数据表
505
507
数据库
报告器
统计表
506
508
WEB 服务器

图 5

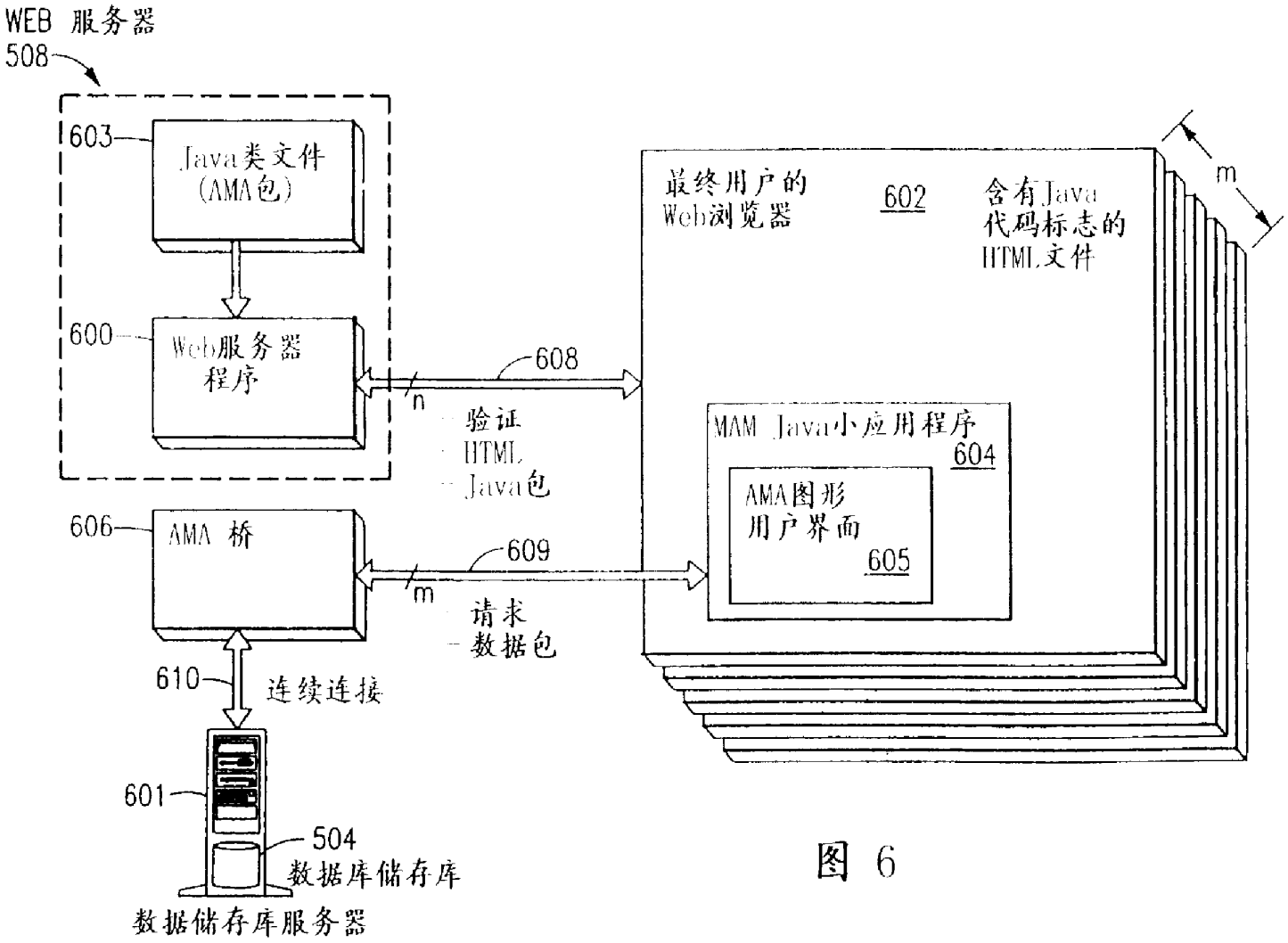
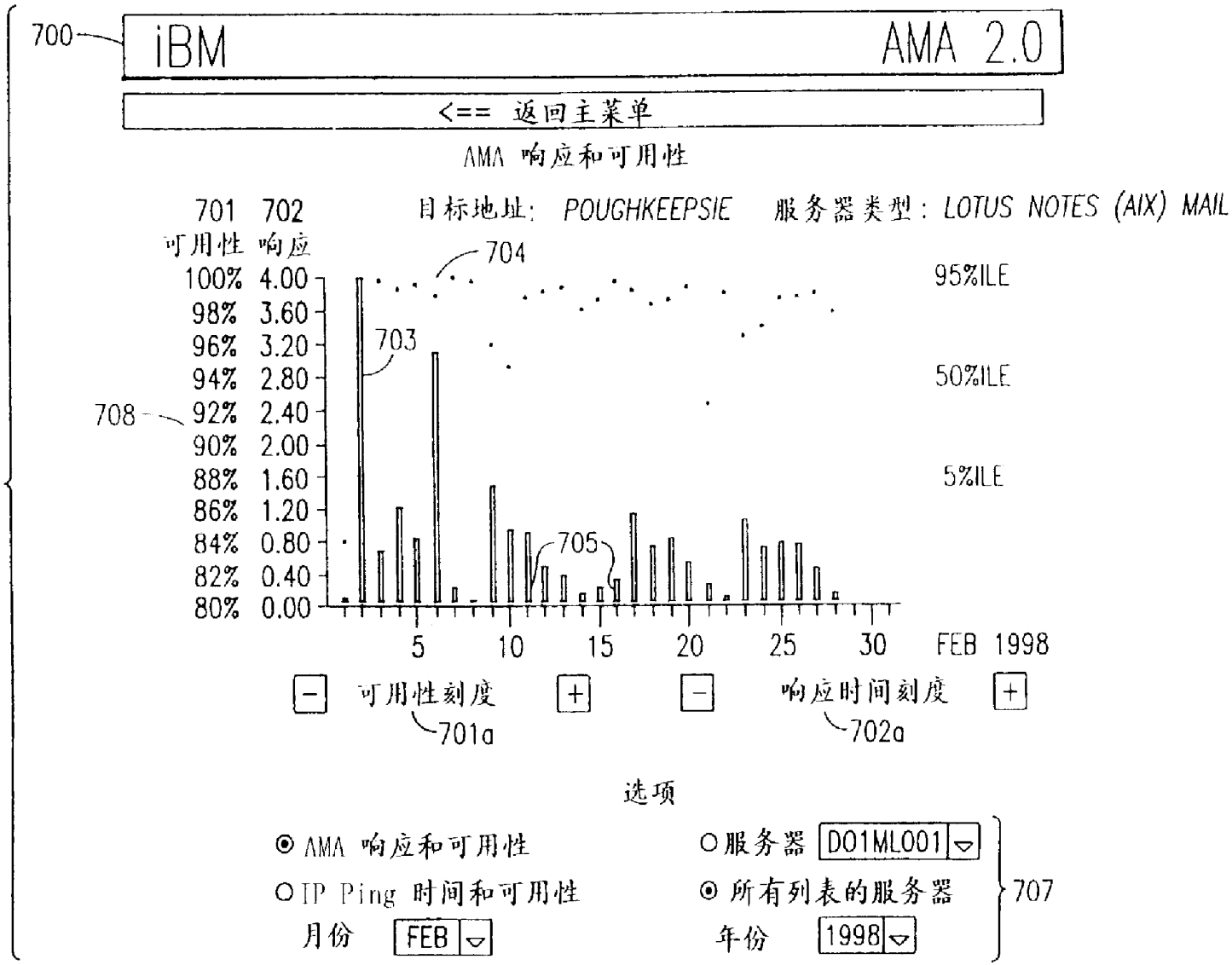

WEB 服务器
508
603
Java类文件
(AMA包)
600
Web服务器
程序
608
n
验证
HTML
Java包
606
AMA 桥
609
m
请求
数据包
610
连续连接
601
504
数据库储存库
数据储存库服务器
最终用户的
Web浏览器
602
含有Java
代码标志的
HTML文件
m
MAM Java小应用程序
604
AMA图形
用户界面
605

图 6

图 7


700
iBM
AMA 2.0
<== 返回主菜单
AMA 响应和可用性
701 702
目标地址: POUGHKEEPSIE 服务器类型: LOTUS NOTES (AIX) MAIL
可用性 响应
100% 4.00
98% 3.60
96% 3.20
94% 2.80
92% 2.40
90% 2.00
88% 1.60
86% 1.20
84% 0.80
82% 0.40
80% 0.00
708
704
703
705
95%ILE
50%ILE
5%ILE
5 10 15 20 25 30 FEB 1998
可用性刻度
701a
响应时间刻度
702a
选项
AMA 响应和可用性
IP Ping 时间和可用性
月份 FEB
服务器 DO1ML001
所有列表的服务器
年份 1998
707

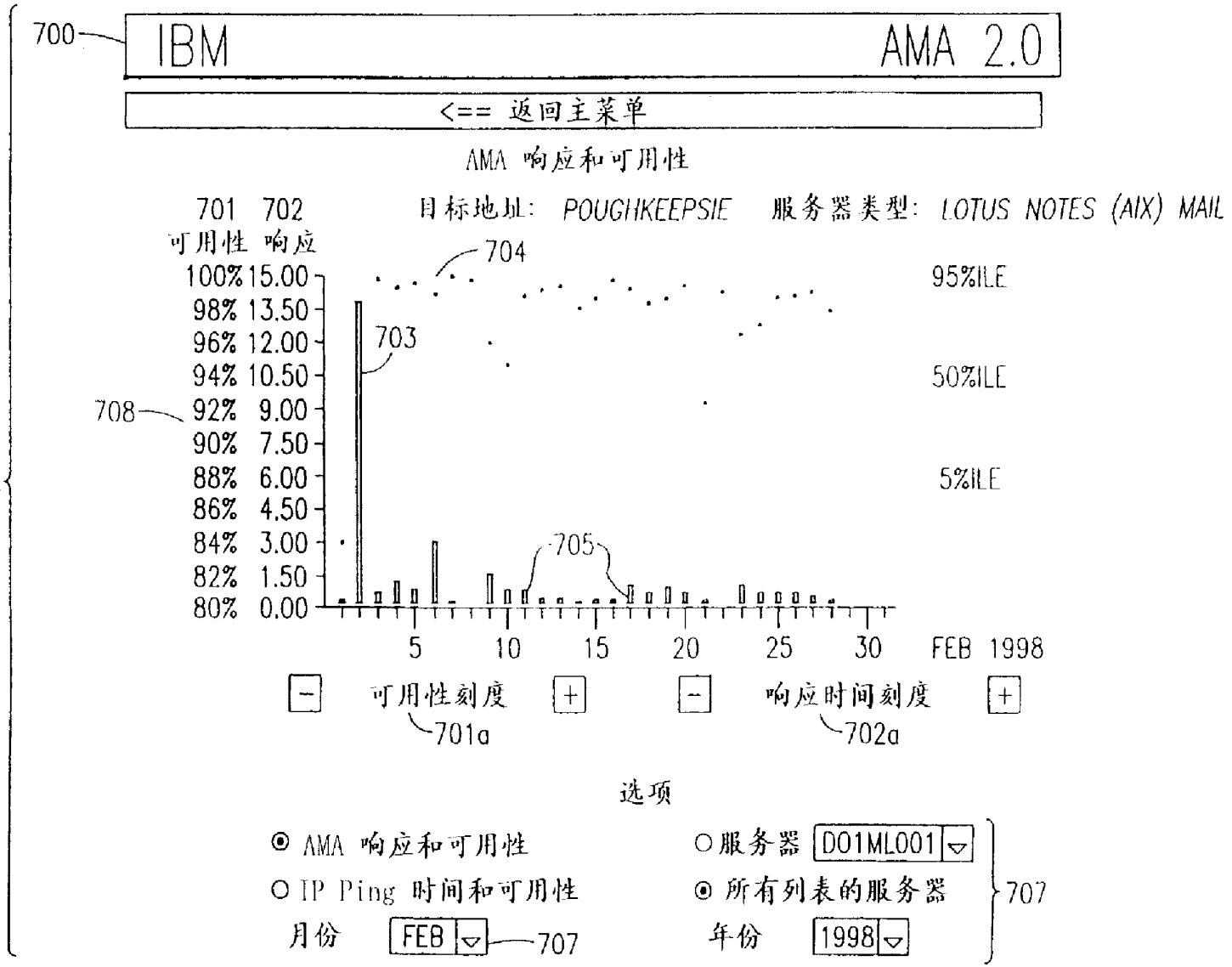

700
IBM
AMA 2.0
<== 返回主菜单
AMA 响应和可用性
701 702
可用性 响应
目标地址: POUGHKEEPSIE
服务器类型: LOTUS NOTES (AIX) MAIL
704
703
708
705
100% 15.00
98% 13.50
96% 12.00
94% 10.50
92% 9.00
90% 7.50
88% 6.00
86% 4.50
84% 3.00
82% 1.50
80% 0.00
95%ILE
50%ILE
5%ILE
5 10 15 20 25 30 FEB 1998
可用性刻度
701a
响应时间刻度
702a
选项
AMA 响应和可用性
IP Ping 时间和可用性
月份 FEB
707
服务器 D01ML001
所有列表的服务器
年份 1998
707

图 7a

图 8

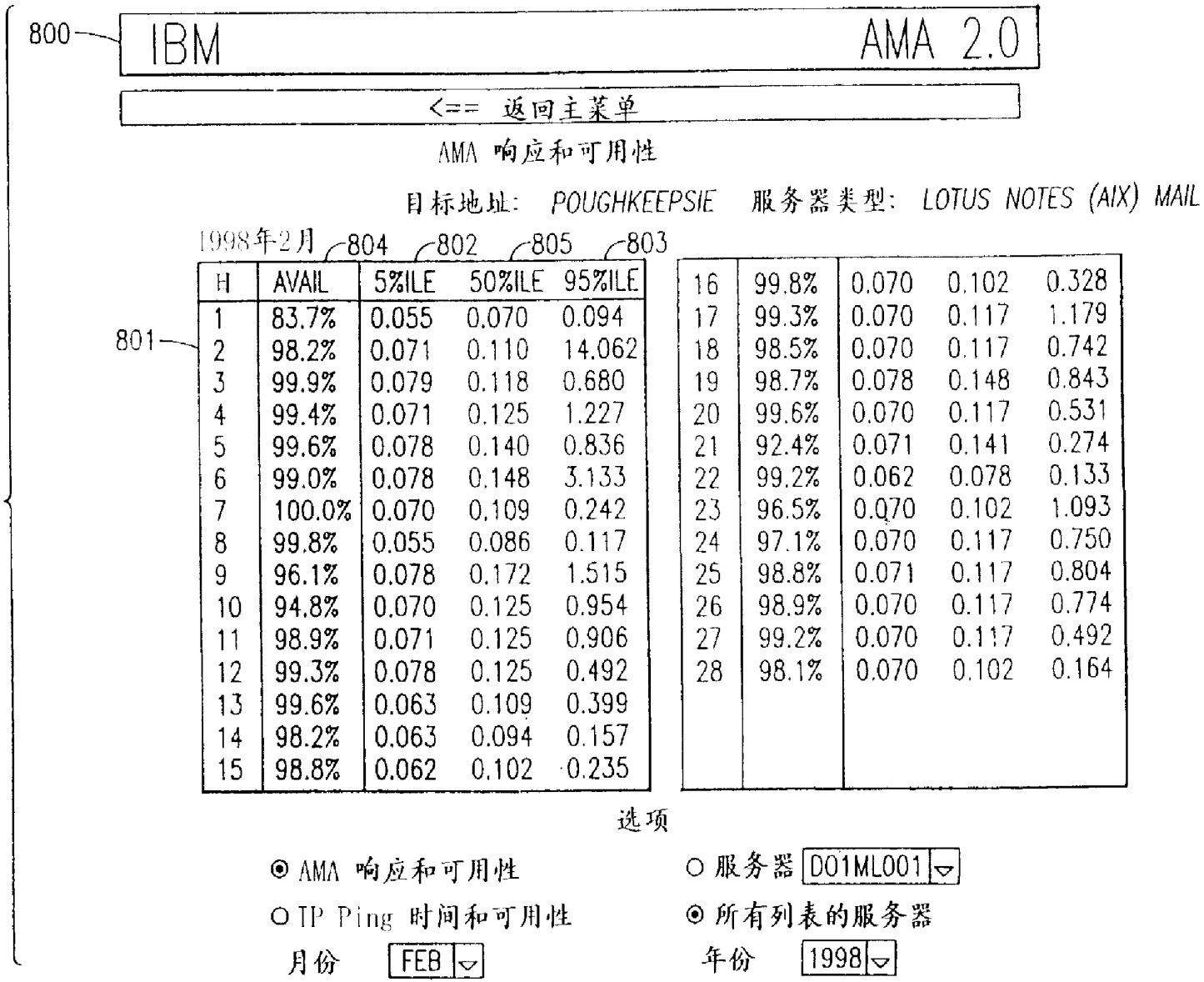
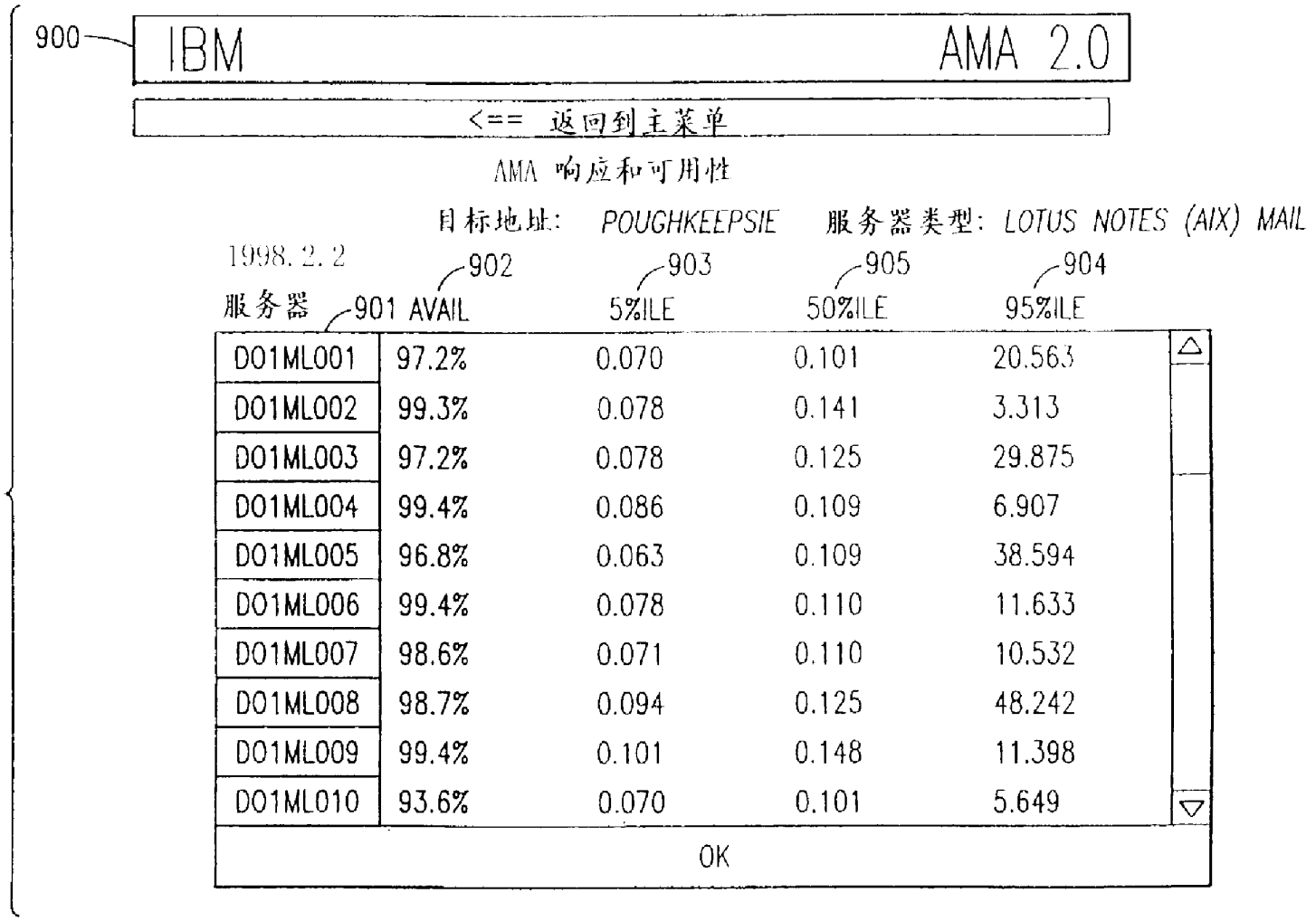

800
IBM
AMA 2.0
<== 返回主菜单
AMA 响应和可用性
目标地址: POUGHKEEPSIE 服务器类型: LOTUS NOTES (AIX) MAIL
1998年2月
804
802
805
803
801
H AVAIL 5%ILE 50%ILE 95%ILE
1 83.7% 0.055 0.070 0.094
2 98.2% 0.071 0.110 14.062
3 99.9% 0.079 0.118 0.680
4 99.4% 0.071 0.125 1.227
5 99.6% 0.078 0.140 0.836
6 99.0% 0.078 0.148 3.133
7 100.0% 0.070 0.109 0.242
8 99.8% 0.055 0.086 0.117
9 96.1% 0.078 0.172 1.515
10 94.8% 0.070 0.125 0.954
11 98.9% 0.071 0.125 0.906
12 99.3% 0.078 0.125 0.492
13 99.6% 0.063 0.109 0.399
14 98.2% 0.063 0.094 0.157
15 98.8% 0.062 0.102 0.235
16 99.8% 0.070 0.102 0.328
17 99.3% 0.070 0.117 1.179
18 98.5% 0.070 0.117 0.742
19 98.7% 0.078 0.148 0.843
20 99.6% 0.070 0.117 0.531
21 92.4% 0.071 0.141 0.274
22 99.2% 0.062 0.078 0.133
23 96.5% 0.070 0.102 1.093
24 97.1% 0.070 0.117 0.750
25 98.8% 0.071 0.117 0.804
26 98.9% 0.070 0.117 0.774
27 99.2% 0.070 0.117 0.492
28 98.1% 0.070 0.102 0.164
选项
◉ AMA 响应和可用性
○ TP Ping 时间和可用性
月份 FEB
○ 服务器 D01ML001
◉ 所有列表的服务器
年份 1998

图 9


900
IBM
AMA 2.0
<== 返回到主菜单
AMA 响应和可用性
目标地址: POUGHKEEPSIE
服务器类型: LOTUS NOTES (AIX) MAIL
1998. 2. 2
902
903
905
904
服务器
901 AVAIL
5%ILE
50%ILE
95%ILE
D01ML001 97.2% 0.070 0.101 20.563
D01ML002 99.3% 0.078 0.141 3.313
D01ML003 97.2% 0.078 0.125 29.875
D01ML004 99.4% 0.086 0.109 6.907
D01ML005 96.8% 0.063 0.109 38.594
D01ML006 99.4% 0.078 0.110 11.633
D01ML007 98.6% 0.071 0.110 10.532
D01ML008 98.7% 0.094 0.125 48.242
D01ML009 99.4% 0.101 0.148 11.398
D01ML010 93.6% 0.070 0.101 5.649
OK

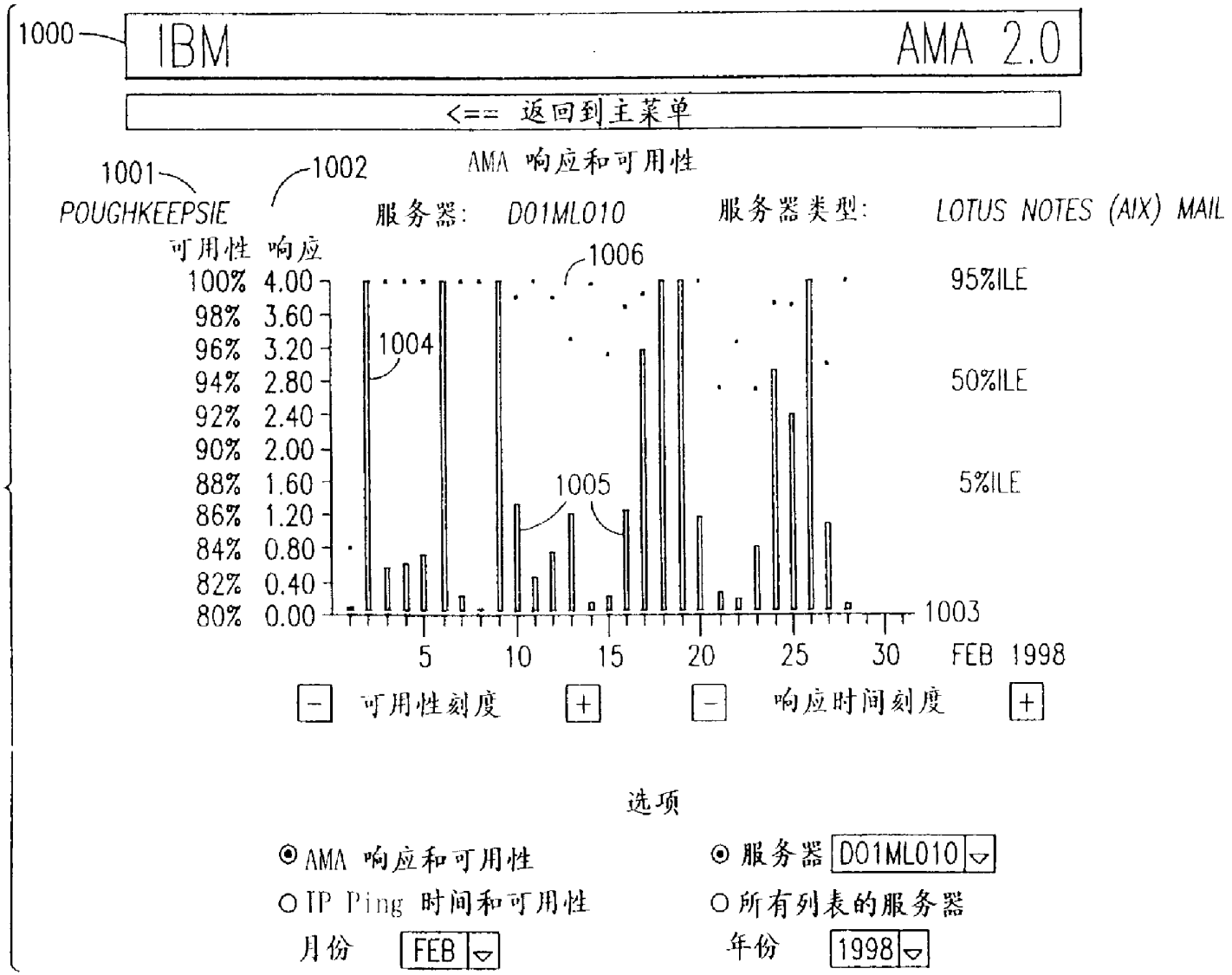

1000
IBM
AMA 2.0
<== 返回到主菜单
AMA 响应和可用性
1001
1002
POUGHKEEPSIE
服务器: D01ML010
服务器类型: LOTUS NOTES (AIX) MAIL
可用性 响应
100% 4.00
98% 3.60
96% 3.20
94% 2.80
92% 2.40
90% 2.00
88% 1.60
86% 1.20
84% 0.80
82% 0.40
80% 0.00
1004
1005
1006
1003
95%ILE
50%ILE
5%ILE
5
10
15
20
25
30
FEB 1998
可用性刻度
响应时间刻度
选项
AMA 响应和可用性
IP Ping 时间和可用性
服务器 D01ML010
所有列表的服务器
月份 FEB
年份 1998

图 10

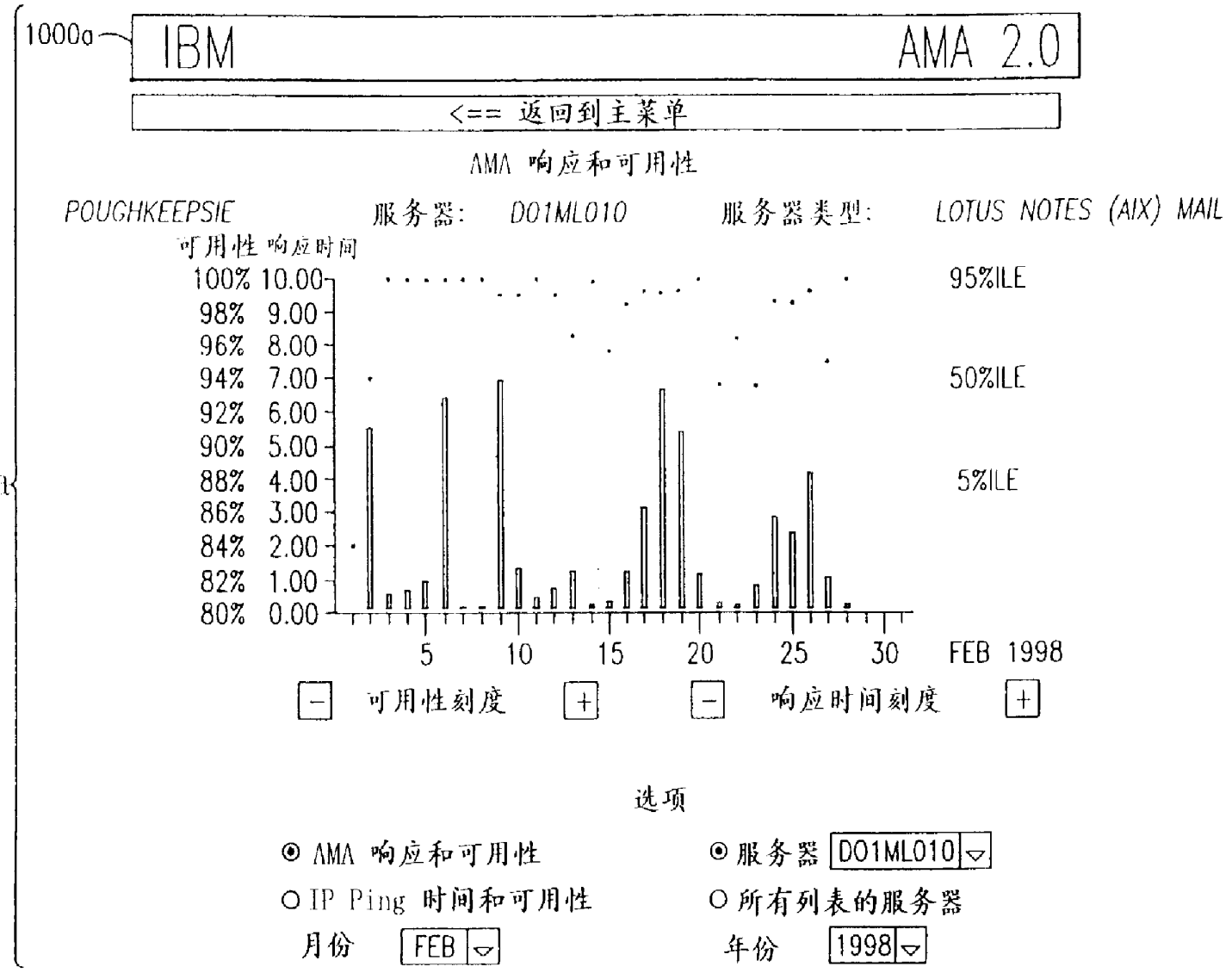

1000a
IBM
AMA 2.0
<== 返回到主菜单
AMA 响应和可用性
POUGHKEEPSIE
服务器: D01ML010
服务器类型: LOTUS NOTES (AIX) MAIL
可用性 响应时间
100% 10.00
98% 9.00
96% 8.00
94% 7.00
92% 6.00
90% 5.00
88% 4.00
86% 3.00
84% 2.00
82% 1.00
80% 0.00
95%ILE
50%ILE
5%ILE
5
10
15
20
25
30
FEB 1998
可用性刻度
响应时间刻度
选项
AMA 响应和可用性
IP Ping 时间和可用性
服务器 D01ML010
所有列表的服务器
月份 FEB
年份 1998

图 10a

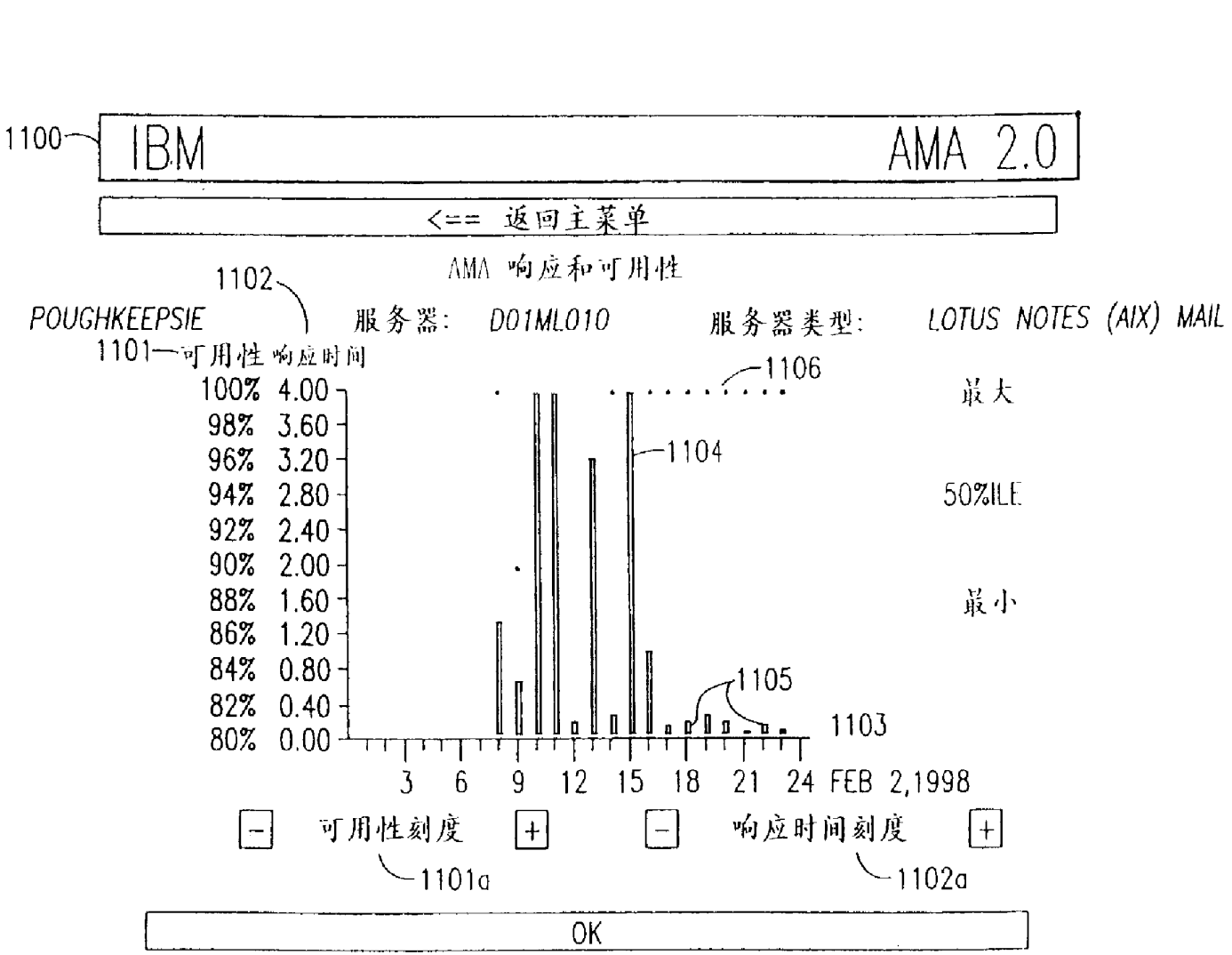

1100
IBM
AMA 2.0
<== 返回主菜单
AMA 响应和可用性
1102
POUGHKEEPSIE
服务器: D01ML010
服务器类型: LOTUS NOTES (AIX) MAIL
1101 可用性 响应时间
100% 4.00
98% 3.60
96% 3.20
94% 2.80
92% 2.40
90% 2.00
88% 1.60
86% 1.20
84% 0.80
82% 0.40
80% 0.00
1106
1104
1105
1103
最大
50%ILE
最小
3 6 9 12 15 18 21 24 FEB 2,1998
可用性刻度
响应时间刻度
1101a
1102a
OK

图 11

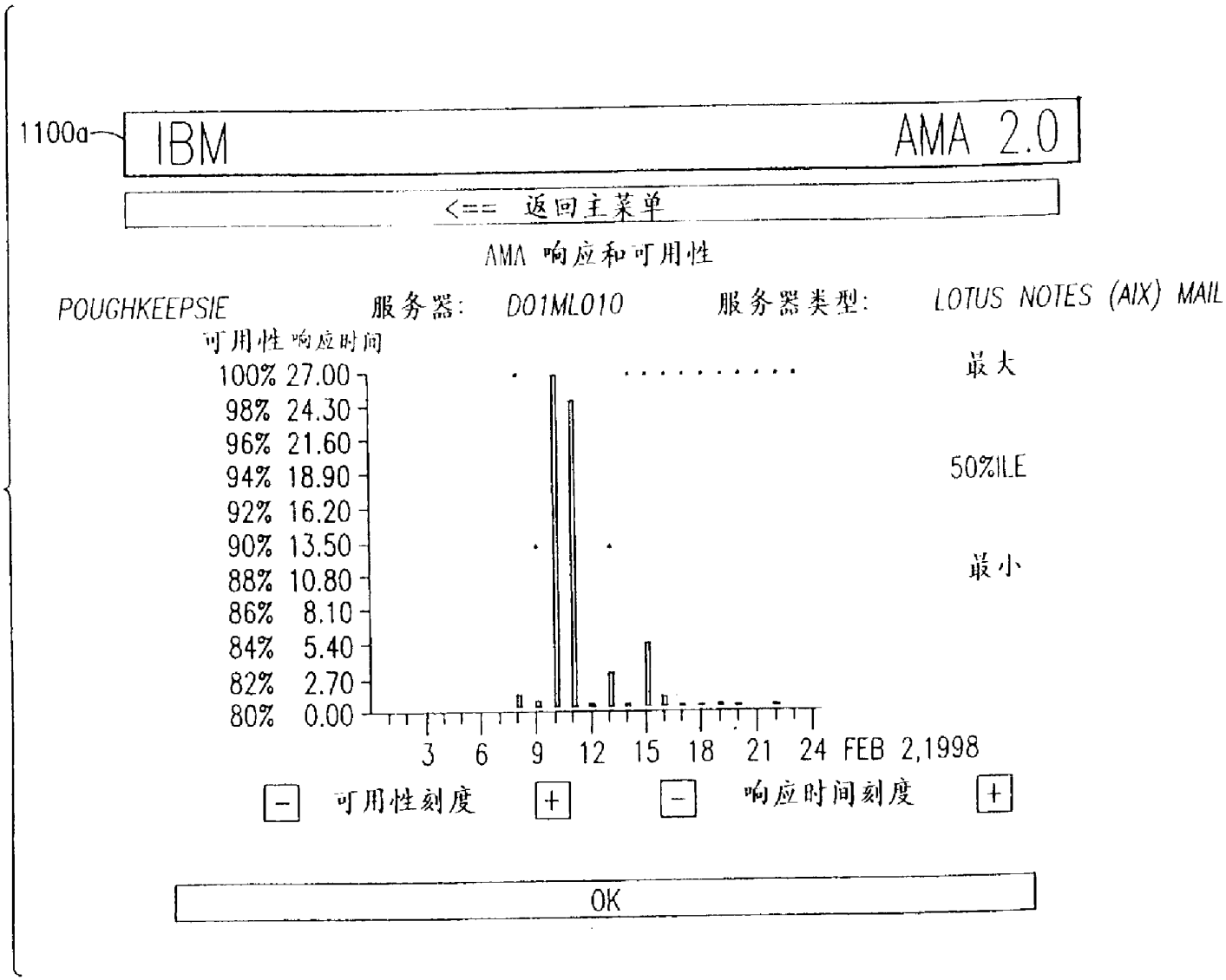
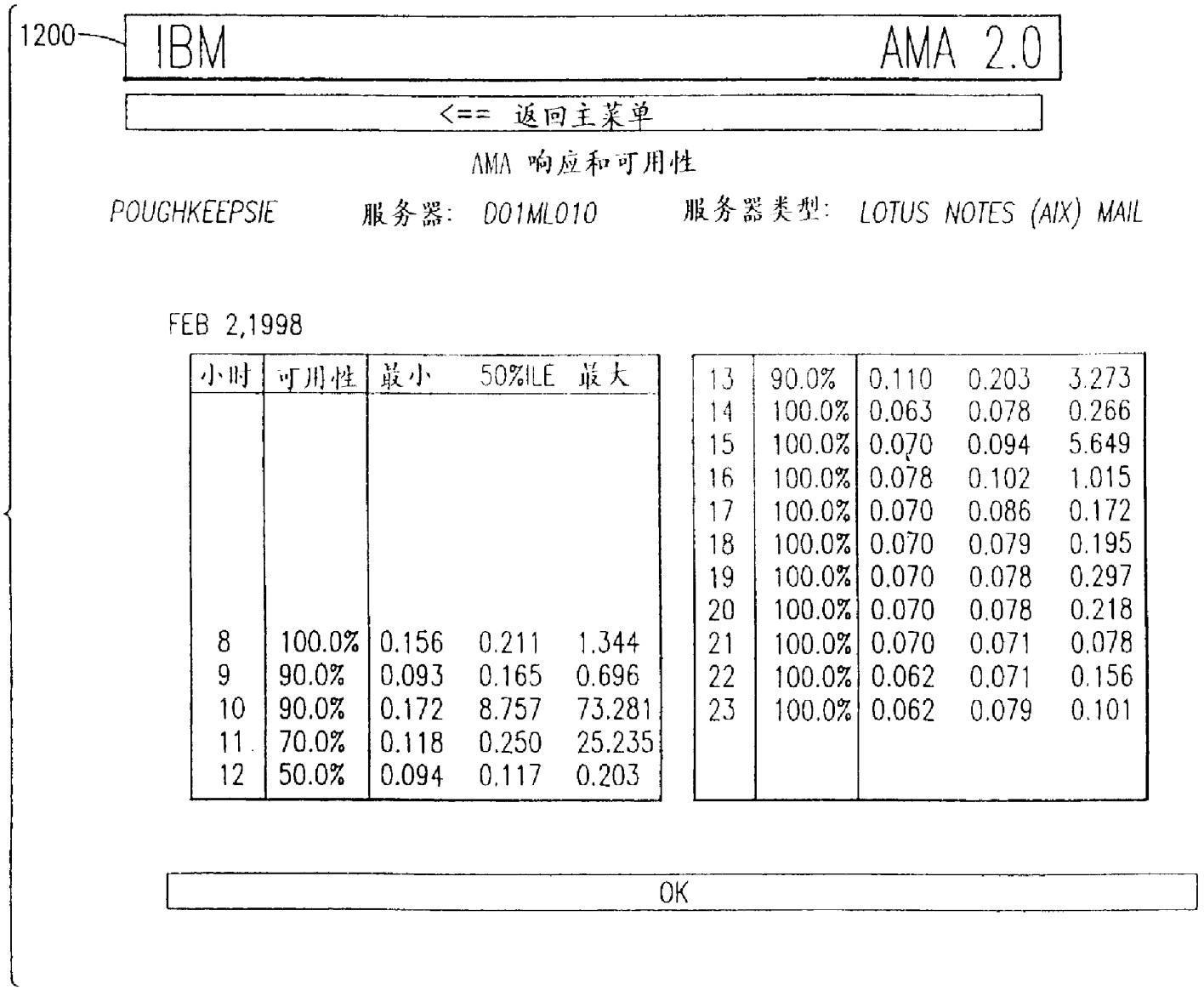

1100a
IBM
AMA 2.0
<== 返回主菜单
AMA 响应和可用性
POUGHKEEPSIE
服务器: D01ML010
服务器类型: LOTUS NOTES (AIX) MAIL
可用性 响应时间
100% 27.00
98% 24.30
96% 21.60
94% 18.90
92% 16.20
90% 13.50
88% 10.80
86% 8.10
84% 5.40
82% 2.70
80% 0.00
最大
50%ILE
最小
3 6 9 12 15 18 21 24 FEB 2,1998
- 可用性刻度 +
- 响应时间刻度 +
OK

图 11a

图 12


1200
IBM
AMA 2.0
<== 返回主菜单
AMA 响应和可用性
POUGHKEEPSIE
服务器: D01ML010
服务器类型: LOTUS NOTES (AIX) MAIL
FEB 2,1998
小时 可用性 最小 50%ILE 最大
8 100.0% 0.156 0.211 1.344
9 90.0% 0.093 0.165 0.696
10 90.0% 0.172 8.757 73.281
11 70.0% 0.118 0.250 25.235
12 50.0% 0.094 0.117 0.203
13 90.0% 0.110 0.203 3.273
14 100.0% 0.063 0.078 0.266
15 100.0% 0.070 0.094 5.649
16 100.0% 0.078 0.102 1.015
17 100.0% 0.070 0.086 0.172
18 100.0% 0.070 0.079 0.195
19 100.0% 0.070 0.078 0.297
20 100.0% 0.070 0.078 0.218
21 100.0% 0.070 0.071 0.078
22 100.0% 0.062 0.071 0.156
23 100.0% 0.062 0.079 0.101
OK

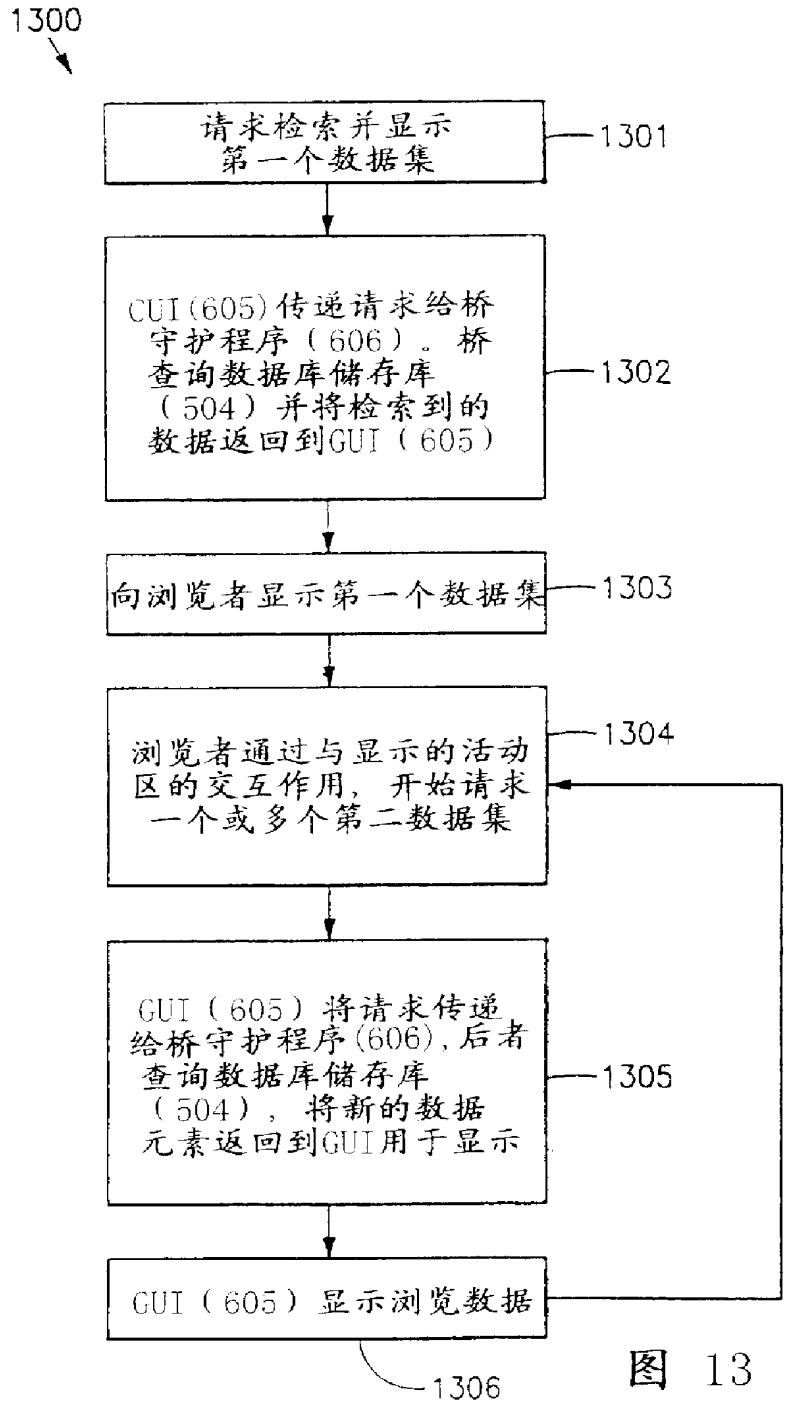

1300
请求检索并显示第一个数据集
1301
CUI(605)传递请求给桥守护程序（606）。桥查询数据库储存库（504）并将检索到的数据返回到GUI（605）
1302
向浏览者显示第一个数据集
1303
浏览者通过与显示的活动区的交互作用，开始请求一个或多个第二数据集
1304
GUI（605）将请求传递给桥守护程序(606)，后者查询数据库储存库（504），将新的数据元素返回到GUI用于显示
1305
GUI（605）显示浏览数据
1306

图 13

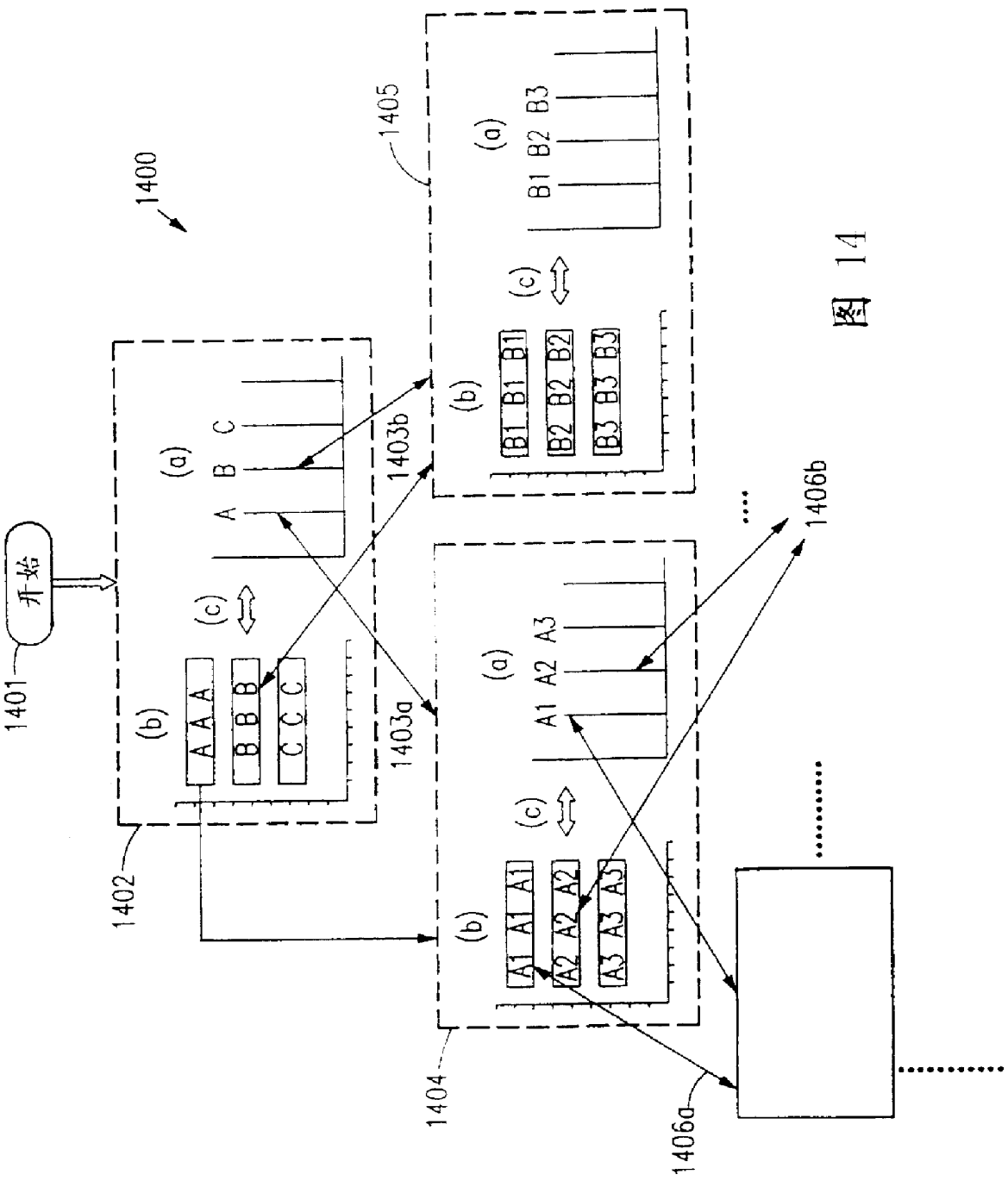

1400
1401
开始
1402
(a)
A B C
(b)
A A A
B B B
C C C
(c)
1403a
1403b
1404
(a)
A1 A2 A3
(b)
A1 A1 A1
A2 A2 A2
A3 A3 A3
(c)
1405
(a)
B1 B2 B3
(b)
B1 B1 B1
B2 B2 B2
B3 B3 B3
(c)
1406a
1406b

图 14

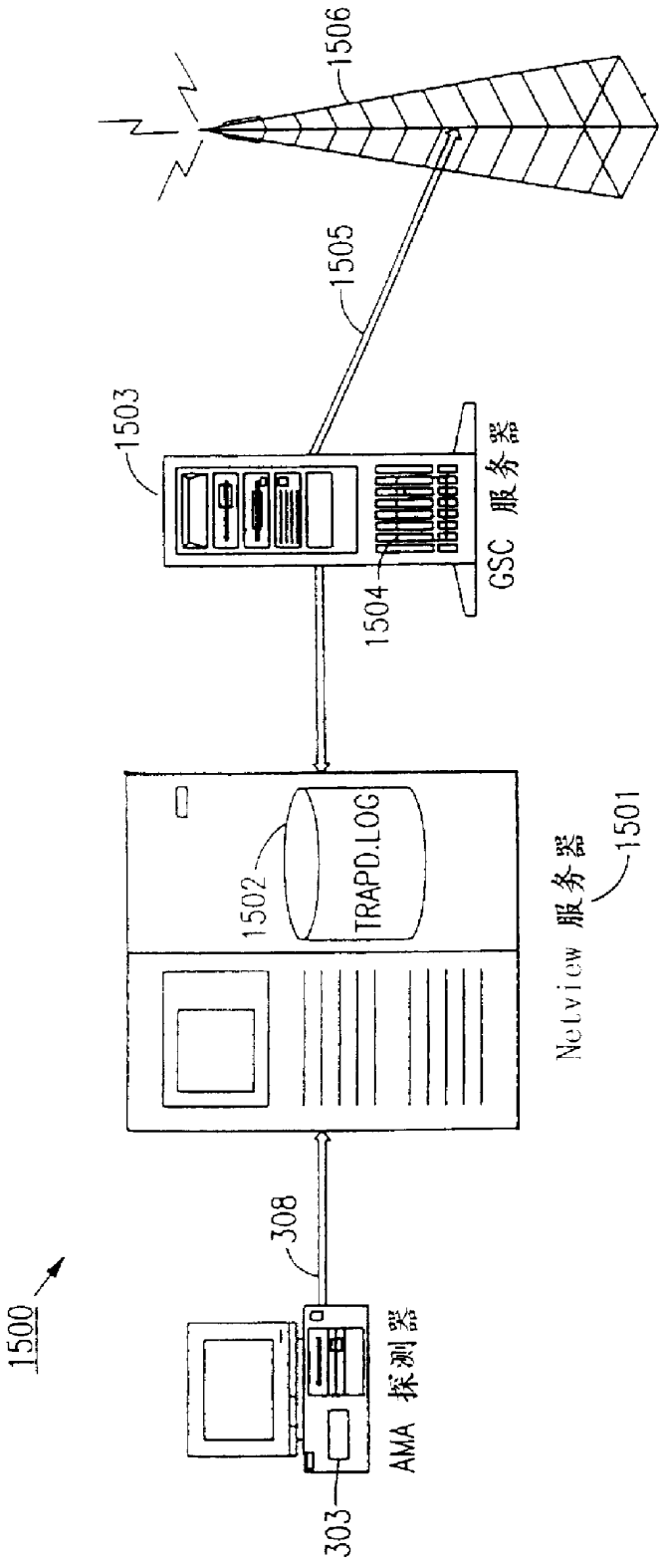

1500
303
AMA 探测器
308
Netview 服务器
1501
1502
TRAPD.LOG
1503
1504
GSC 服务器
1505
1506

图 15